Classification of Robotic Arm Movement using SVM and Naïve Bayes Classifiers

Asha Vijayan*
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam, Kerala
ashavijayan@am.amrita.edu

Chaitanya Medini*
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam, Kerala
krishnachaitanya@am.amrita.edu

Hareesh Singanamala*
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam, Kerala
hareeshsinganamala@am.amrita.edu

Chaitanya Nutakki*
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam, Kerala
chaithanyakumar@am.amrita.edu

Bipin Nair
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham
Clappana (P.O.), Kollam, Kerala
bipin@amrita.edu

Shyam Diwakar#
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University),
Clappana (P.O.), Kollam, Kerala
shyam@amrita.edu

*Authors contributed equally, # Corresponding author

Abstract—Target-oriented approaches have been commonly used in robotics. In 3D space, movement of a robotic arm depends on the target position which can either follow a forward or inverse kinematics approach to reach the target. Predicting the movement of a robotic arm requires prior learning through methods such as transformation matrices or other machine learning techniques. In this paper, we built an online robotic arm to extract movement datasets and have used machine learning algorithms to predict robotic arm articulation. For efficient training, small training datasets were used for learning purpose. Classification is used as a scheme to replace prediction-correction approach and to test whether the method can function as a replacement of usual forward kinematics schemes or predictor-corrector methods in directing a remotely controlled robotic articulator. This study reports significant classification accuracy and efficiency on real and synthetic datasets generated by the device. The study also suggests linear SVM and Naïve Bayes algorithms as alternatives for computational intensive learning schemes while predicting articulator movement in laboratory environments.

***Index Terms*—Robotic Articulator, Movement, Machine Learning, experimental dataset, Naïve Bayes, SVM.**

I. Introduction

Robotic interaction with the real world environment has gained prominent importance in recent years in various fields and applications [1]. Modern robotic manipulators are constructed using a set of joints which are combined together using rigid links [2]. These manipulators are developed to do specialized tasks like examining different parameters of an object [3], reaching the target with an optimal feedback control [4], etc. One of the main areas of research in humanoid robotics has been on the identification of objects based on spatial relationship [5]. Different robotic tasks related to object detection require large datasets to build good object classifiers [6]. Previous studies [7] have shown that the classifiers like support vector machine (SVM), Decision tree(C 4.5 algorithm) and Naïve Bayes classifiers can be used to predict surface textures with the help of opto-tactile sensors. Here, in this paper, we used linear SVM, nonlinear SVM and Naïve Bayes classifiers on two datasets which were generated using an online virtual lab simulator [8, 9] for predicting movement of the articulator. Remotely-triggered experiments on robotic articulator are available online (http://amrita.edu/virtuallabs) and were used to generate the dataset on which these algorithms were tested.

The goal of this study was to substitute a bio-inspired trajectory targeting approach (e.g. Kalman filter (KF) trained Multilayer Perceptron (MLP), unpublished data), which has been used as a trajectory-tracking model, using a classifier. The drawback of predictor-corrector methods is that the computational cost was high when the dataset was nonlinear [21]. To substitute and complement such predictor- corrector models with simpler methods, we suggest a linear hyperplane or probability-based stochastic classifier through this study.

In order to test and validate the hypothesis of substituting with simpler classification schemes, two types of classifiers on the datasets: SVM with linear and non-linear kernels and Naïve Bayes classifier were used. Linear SVMs are popular for their ability to classify large datasets in simple and fast way [10, 11]. Naïve Bayes (NB) classifier has been widely used as a supervised learning technique and is well-known for its ability to perform better than non-probabilistic

978-1-4799-0048-0/13/$31.00 ©2013 IEEE

classification approaches such as neural networks and decision trees [18, 19, 20]. NB classification assumes conditional independence and Bayes theorem to estimate probability values of success/failure of an event [12, 13].

In this study, the robotic hardware was developed using a low-cost approach and calibrations were done to model the articulator as a simpler version of a prosthetic device. The classification algorithms were tested on the normalized dataset to classify a particular set of given attributes (motor values and end-effector coordinates) based on their ability to train the articulator to reach a 'known' target. Normalized dataset contains both training and test data wherein test data was a subset of the training data whose class label need to be predicted. A polynomial kernel in the non-linear SVM algorithm was also used. Alongside with the datasets obtained by experimentation, the classifiers were pre-tested on standard datasets such as weather dataset [22] and compared it to other algorithmic methods.

II. METHODS

A. Robotic Arm

A robotic articulator was constructed using rigid links which have 5 degrees of freedom (DOF) and a grasper with 6-DOF (see Figure 1A, online experiment freely available at http://amrita.vlab.co.in/index.php?sub=3&brch=257&sim=1458&cnt=3171). Each link constitutes a servo motor with a torque of 17 kg-cm at 6 Volts. A microcontroller (Figure 1B) was programmed to generate PWM (Pulse Width Modulation) signals with a time period of 20ms and a duty cycle varying from 1ms to 2ms for controlling the motors of the articulator. Microcontroller was interfaced through a serial port (RS-232) to the PC from which the desired motor angles will be transferred to the controller (Figure 2). User may perform the experiment in two modes: firstly, forward kinematics: where the user provided motor angle and obtained the end-effector and secondly, inverse kinematics: where the user provided the end-effector and obtained the individual motor angles which were given as inputs to the microcontroller. With these motor angles, the microcontroller generated control signal for each motor.

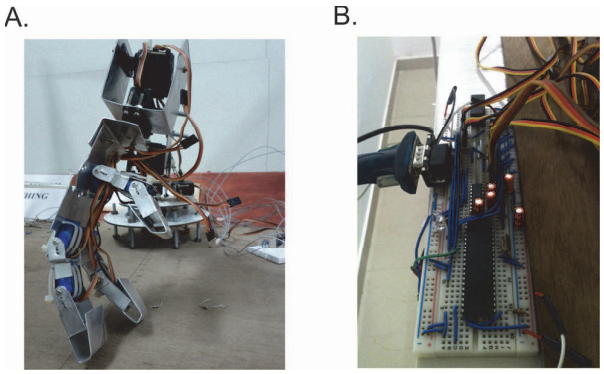

Fig. 1. Setup. A. Robotic Articulator with 11 DOF. B. Microcontroller Setup

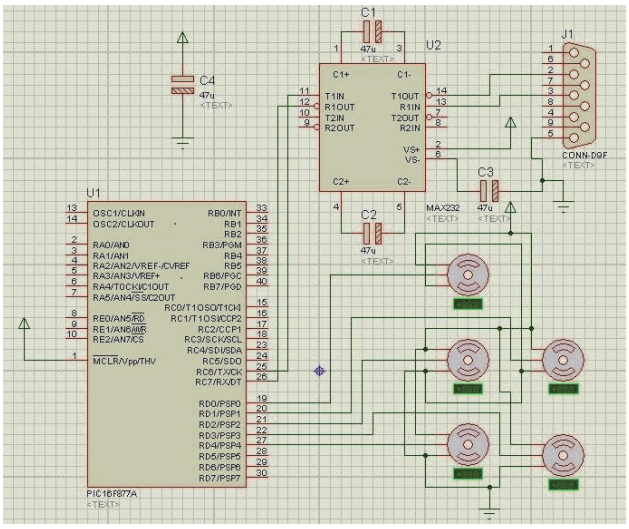

Fig. 2. Schematics of the control circuit

This setup (with a gripper instead of a grasper) can be remotely-triggered via Amrita virtual labs and is part of the National mission project on Education through ICT of the Government of India [17].

B. Generating Datasets

The datasets were obtained by two methods, simulation (synthetic dataset) using a kinematics approach [23] while the other was from the online virtual lab robotic tool using Cartesian coordinate measurements (experimental dataset). Datasets containing the motor angles of the 6 motors and their corresponding end-effector co-ordinates were generated using the experimental setup shown in Figure 1. The class label (y) determined whether supplied values indicated if the arm reached the target object. If the target was reached given input pulses, class label was assigned as +1, otherwise the target was assigned as -1. The target was chosen by first simulating the motor values and estimating an end-effector coordinate which was cross-validated on the experimental setup. For classification purposes, the motor values of the arm alone (5+1 motor values) were taken and values from the grasper were ignored. The mid-point of the grasper was taken as the end-effector coordinates (generated by both methods) used for classification.

Min-Max method [15] was used to normalize the dataset (see Eq. 1).

$$Norm\_data = \left(\frac{data - \min(data)}{\max(data) - \min(data)}\right) (1)$$

In Eq. 1, "data" refers to the element of an attribute being normalized. "min(data)" refers to the element which has least value in an attribute (motor values and end effector coordinates) for all instances and "max(data)" refers to the element which has high value of the same attribute. "Norm_data" contains the normalized value.

For training and test purposes, both datasets were independently divided into training set and testing set using percentage-split method [16] with a split ratio of 66% (34%

was taken as test data). Training data included the class label. While test data does not include a class label and the class label was predicted by the algorithms. Both datasets consisted of 6 motor values and 3 end-effector coordinate values (9 attributes, see Figure 3). A sample 3D plot with the attributes of 4 instances (in red) are shown in Table 1. Linear SVM (LSVM) and Naïve Bayes implementations were used on both datasets.

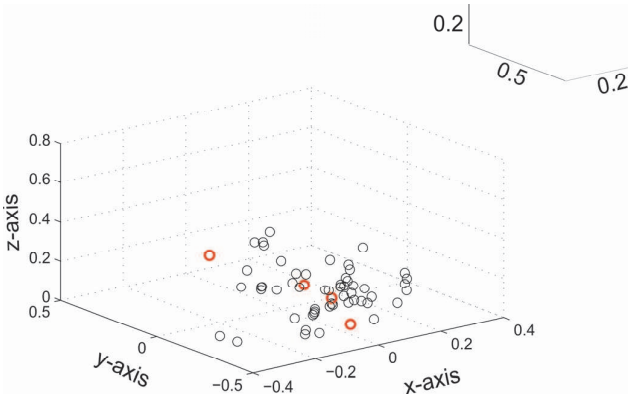

Fig 3. 3D plot of the synthetic dataset

C. Classification

Along with linear kernel in SVM, polynomial kernel was used as non-linear kernel for the dataset. Formulation is as follows. A is normalized data obtained through Eq. 1, polynomial kernel (polyA) is represented as in Eq. 2.

$$polyA = (AA^T + 1)^4 \qquad (2)$$

Table 1. Attribute values of the selected instances from synthetic dataset

Motor 1	**Motor 2**	**Motor 3**	**Motor 4**	**Motor 5**	**Motor 6**	**X**	**Y**	**Z**	**Class label(y)**
120	100	60	120	48	91	-0.0362	0.31842	0.16856	-1
128	67	67	158	112	74	-0.0332	-0.36114	0.05747	-1
99	100	40	160	98	80	-0.1004	-0.27861	0.25965	1
120	100	60	140	20	67	-0.0371	-0.31851	0.18325	1

The obtained polynomial matrix (2) was substituted into the algorithm and thus predicted the class labels for the test data. Consider Bayesian approach, when joint probability distribution is considered for 3 random variables case (A_1, A_2 and V)

$$p(A_1, A_2|V) = p(A_1|V) * p(A_2|V) \qquad (3)$$

$$p(A_1, A_2, \dots, A_k|V) = \pi_{i=1}^{k} p(A_i|V) \qquad (4)$$

In Naïve Bayes, each instance x is expressed as a conjugation of attribute values (A_1, $A_2 \dots A_n$); x=<A_1, $A_2 \dots A_n$>. The target function f(x) € V like V= {+,-}. In our implementation, A_1, A_2, A_6 are motor values, A_7, A_8, A_9 are the XYZ coordinates of the end effector and V is the class label with +1 or -1. The Naïve Bayes classifier is of the form.

$$V_{NB} = \underset{V_j \in V}{\arg max} [\![\prod_{i=1}^{n} P(a_i|v_j)]\!] P(v_j) \qquad (5)$$

The algorithmic implementation for SVM and NB is listed below.

Algorithm for linear SVM [11]:

1. Training data X={x_1, $x_2 \dots x_n$} with class label y € {-1, 1} was provided as input.

2. X along with a column vector (e) containing ones stored in a matrix V. V=[X, -e]

3. V was then multiplied with diagonal matrix (D) containing class labels. H=D*V

4. H was multiplied with its diagonal matrix (H^T) and stored. Let this matrix be tranH=H*H^T.

5. Identity matrix (I) was generated with size of tranH and divided with v which has value of 0.1, let this matrix be M. [I/v]

6. M was added with tranH matrix and stored in N. Inverse of N is taken. N=(I/v+tranH)$^{-1}$

7. Obtain the final matrix using the below equation.

$$u = v * \left(I - H * \left(\frac{I}{v} + H^T H \right)^{-1} * H^T \right) * e$$

8. Finally, w and γ values were obtained. And used them in the equation w^T*x-γto classify the test dataset.

Algorithm for Naïve Bayes [12]:

1. Setting the data

a) Each instance consists of n attributes X: {x_1, x2…x_n).

b) Each of the instance was classified into different classes, C_j where j=1, 2…m.

2. Train the dataset

a) Find the probabilities p(C_j).

b) Using conditional independence, calculate

$$p(X|C_j) = \pi_{i=1}^{k} p(x_k|C_j)$$

3. Predict test data

a) $p(C_j|X) = \frac{p(X|C_j) * p(C_j)}{p(X)}$

b) Maximize $p(C_j|X) = p(X|C_j) * p(C_j)$

4. The error is estimated as follows

a) error = (sum of misclassified data))/Total number of data_points *100

D. Validation

For validation of the algorithms implemented, the weather dataset [16] was used and obtained 78% accuracy for linear SVM, 65% for nonlinear SVM (NLSVM) and 60% for Naïve Bayes. These results were crosschecked with other implementations of same algorithms in WEKA (Waikato Environment for Knowledge Analysis) [16].

III. RESULTS

The arm, developed with 5+6 joints, was able to grasp objects of around 100g (input voltage is 7.5V for each motor) at a given position and has been tested repeatedly to check if there were errors in 'reaching' a given position. If the object weight exceeded beyond 120g (given 7.5V as input voltage to motors), the arm started to stagger. Forward kinematics implementations allowed generating data which was similar to the experimental data recorded from the device.

The percentage-split based training and test datasets were tested on different algorithms using different percentages of split (see Figure 4A, 4B and 4C). Observations suggest that with increase in the percentage of split the training data efficiency decreases whereas test data efficiency increases. Nonlinear SVM showed significantly higher test data efficiency when compared to linear SVM and NB classifiers. We have used 66% split to compare the training and test data efficiency as well as error since the results for different percentages were almost similar (Figure 4).

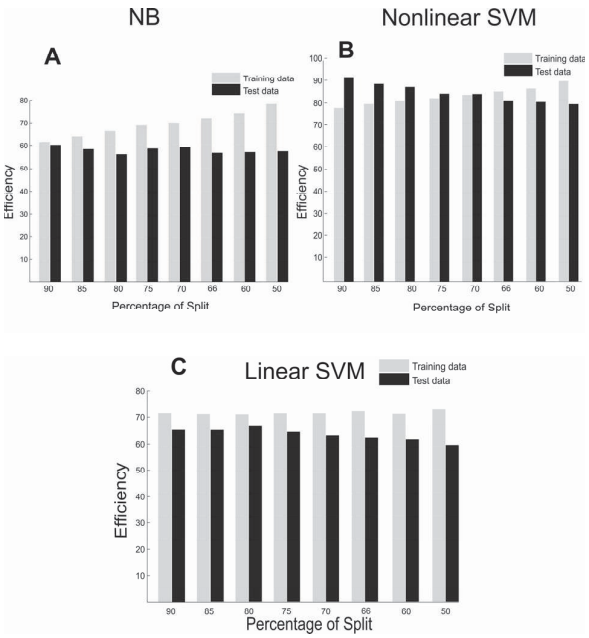

Fig. 4. Classification of robotic datasets. A. Naïve Bayes (NB) classifier efficiency over different percentages of split. B. Nonlinear SVM efficiency. C. Linear SVM efficiency.

Table 2 shows the different efficiencies observed based on the methods used (LSVM, NLSVM or NB classifier) with 66% split. Figure 5A shows the comparison of training and test efficiency with different methods. Figure 5B shows the comparison between training and test error. In both cases (training and test), SVM with polynomial kernel of degree 4 showed very less error (Figure 5B) and gave reasonably higher correct classification (Figure 5A). Linear SVM showed a close accuracy to nonlinear SVM in both cases (training and test), NB classifier showed relatively less percentage of efficiency with training data, however the test data showed improved efficiency suggesting that it can be used as a good classifier.

Table 2. Efficiency in percentages for the dataset with different algorithms

	Testing			Training		
	LSVM	NL SVM	NB	LSVM	NL SVM	NB
Efficiency of the data (%)	62.2	82	56.8	72.1	86	72.1

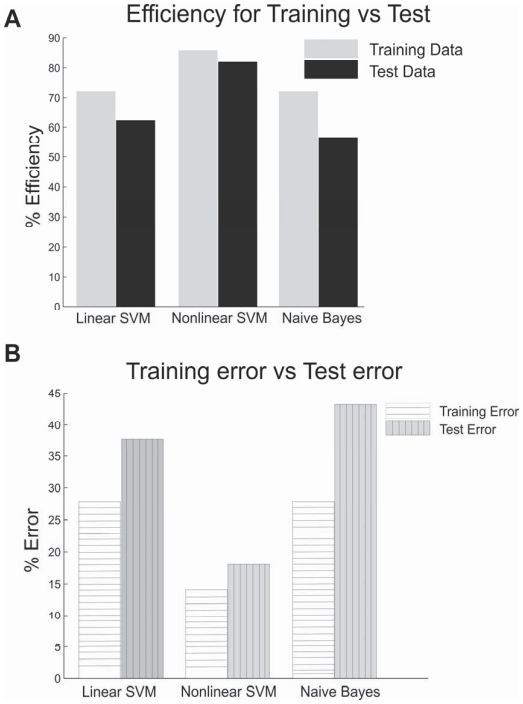

Fig.5. A. Comparison of Training (gray) vs Test (black) efficiencies. B. Comparison of Training vs Test error.

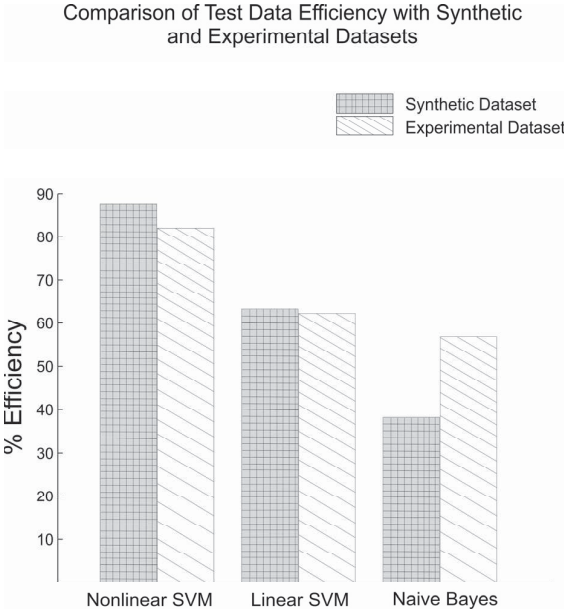

Fig. 6. Comparison of test data efficiency between synthetic and experimental datasets.

Test data efficiency was compared between synthetic and experimental datasets. Classifiers on experimetal dataset showed higher performance in case of linear and nonlinear SVM when compared to the same on synthetic dataset (Figure 6). However the efficiency of NB classifier showed higher perfomance in case of synthetic dataset than on experimental dataset (Figure 6).

The results suggest that linear SVM and NB classifiers can be used to classify small articulator movement datasets with reasonable efficiency with reduced computational cost unlike nonlinear kernels. The simulation time required for the training of synthetic dataset was as follows, 0.741883s for linear SVM, 0.766669s for nonlinear SVM and 0.136662s for NB classifier. Experimental dataset training showed the simulation time as follows, 0.880s for linear SVM, 1.13s for nonlinear SVM and 0.17 s for NB classifier. All simulations were done on Dell T3500 workstation with Ubuntu Linux 13.04 (SMP x86_64 kernel 3.8.0-19-generic) operating system on a Intel Xeon W3670 CPU (frequency of 3.20 GHz) with 8 GB RAM. These observations indicate linear and NB classifiers take less computational time when compared to Kalman filter trained MLP (KF-MLP, data not shown).

IV. Discussion

We have attempted using a machine learning approach rather than detailed bio-inspired approach wherein a linear classifier was used to define the movement prediction in a robotic articulator. The robotic arm was constructed with an idea of achieving generalization over accuracy to reach a given target. We used 70 point dataset in this study and showed significant (82%) classification accuracy. In the experimental dataset, the Denavit-Hartenberg (DH) parameters (D) [23] were generated from the kinematics model on which the forward kinematics was applied to obtain transformation matrix (a combination of rotational and translation matrix (end effector co-ordinates)).In the current approach of classification, the matrix has been replaced with a weight matrix (W) of dimension 9x1 in the linear hyper-plane classifier. The generated datasets were validated through repeatability allowing the robotic online articulator to be used for extracting the datasets related to movement classification studies. The intentional lack of precision in some data points was due to design restrictions of allowing the consideration of the robotic arm as a low cost prosthetic device. For classification purpose, linear SVM proved to be a promising approach rather than nonlinear SVM since nonlinear SVM has higher computational cost. While on the other hand, NB showed higher percentage of error, however the test dataset after training showed relatively less error. Over-learning was prevented by decreasing number of training instances and found that the classification levels did not vary drastically (test efficiency was 87.5% with 40 data points, probably due to overlearning).

Classification results indicated that certain linear classifiers perform with comparable efficiency as nonlinear classifiers thereby allowing us to use simpler implementations without any other kernel methods. Alternative implementation in MATLAB provided similar results indicating that computational cost may be overcome by introducing the appropriate modelling techniques. The intention in keeping the dataset small was to verify whether the classifier could identify the generalization properties exhibited by the kinematics model. Efficiency of the classifier is strongly dependent on the number of training vectors since the size of the training data taken was significantly small (n=70), there will be no overlearning

V. Conclusion And Future Work

With this efficiency, we find that typical predictive-correction model such as KF-trained MLP algorithms which are computationally expensive may be replaced using simpler linear SVM classifiers. We are currently extending this work with an alternative approach of using a spiking neuron network model (CIS-NN, unpublished data) substituting firing rate as a metric to measure trajectory patterns.

Acknowledgment

This work derives direction and ideas from the chancellor of Amrita University, Sri Mata Amritanandamayi Devi. This work is partially supported by grants SR/CSI/49/2010 and SR/CSI/60/2011 from DST, BT/PR5142/MED/30/764/2012 from DBT, Sakshat Virtual labs, NMEICT, Government of India and also by Indo-Italy POC 2012-2013.

References

[1] J. Lee, W. B. Knox, and P. Stone, “Inter-Classifier Feedback for Human-Robot Interaction in a Domestic Setting,” Journal of Physical Agents, Vol.2, pp.41-50, June 2008.

[2] S. Tejomurtula and S. Kak, “Inverse kinematics in robotics using neural networks,” Information Sciences, Vol.116, pp. 147-164, 1999.

[3] H. P. Saal, J. A Ting, and S. Vijayakumar, “Active Estimation of Object Dynamics Parameters with Tactile Sensors,” Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2010), Taiwan, 2010.

[4] D. Mitrovic, S. Nagashima, S Klanke, T. Matsubara and S. Vijayakumar, “Optimal Feedback Control for Anthropomorphic Manipulators,” Proc. IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA, 2010.

[5] E. Ho, T. Komura, S. Ramamoorthy and S. Vijayakumar, “Controlling Humanoid Robots in Topology Coordinates,” Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS 2010), Taiwan, 2010.

[6] Z. Jia, A. Saxena and T. Chen, “Robotic Object Detection: Learning to Improve the Classifiers Using Sparse Graphs for Path Planning,” Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

[7] A. M. Mazid and A. B. M. Shawkat Ali, “Opto-tactile Sensor for Surface Texture Pattern Identification using Support Vector Machine,” 10th Intl. Conf. on Control, Automation, Robotics and Vision Hanoi, Vietnam, 17–20 December 2008

[8] B. Nair, R. Krishnan, N. Nizar, R. Radhamani, K. Rajan, A. Yoosef, G. Sujatha, V. Radhamony, K. Achuthan, and S. Diwakar, “Role of ICT-enabled visualization-oriented virtual laboratories in Universities for enhancing biotechnology education – VALUE initiative: Case study and impacts,” FormaMente, vol. VII, no. 1-2, ISSN 1970-7118, 2012.

[9] S. Diwakar, K. Achuthan, P. Nedungadi and B. Nair, “Enhanced Facilitation of Biotechnology Education in Developing Nations via Virtual Labs: Analysis, Implementation and Case-studies,” International Journal of Computer Theory and Engineering, Vol. 3, no. 1, pp. 1-8, 2011.

[10] L. Ladicky, and P.H.S. Torr, “Locally Linear Support Vector Machines,” Proceedings of the Twenty-Eighth International Conference on Machine Learning, 2011.

[11] C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, Vol 20, pp. 273-297, 1995.

[12] H. Zhang, "The Optimality of Naive Bayes," FLAIRS2004 conference, 2004.

[13] G. H. John and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers,” Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, pp. 338-345, 1995.

[14] J. Wang, and T. Downs, “Improving Support Vector Machine Performance on a Problem in Object Recognition,” Complexity International, Vol. 12, 2005.

[15] J. Han, and M. Kamber, “Data Mining: Concepts and Techniques”, Second edition, 2006, Morgan Kaufmann, USA.

[16] G. Holmes, A. Donkin and I.H. Witten, "Weka: A machine learning workbench," Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia, Retrieved 2007-06-25, 1994.

[17] S. Diwakar, K. Achuthan, P. Nedungadi, “Biotechnology Virtual Labs – Integrating Wet-lab Techniques and Theoretical Learning for Enhanced Learning at Universities, ”Proceedings of the International Conference on Data Storage and Data Engineering (DSDE 2010), Bangalore, Feb 10, 2010.

[18] H. H. Aviles-Arriaga, L. E. Sucar, C. E. Mendoza, B. Vargas, “Visual Recognition of Gestures using Dynamic Naïve Bayesian Classifiers,” Proceedings of the 2003 IEEE international workshop on Robot and Human Interactive Communication, Millbrae, California, USA, 2003.

[19] D. Michie, D. J. Speigel Halter, C. C. Taylor, “Machine Learning, Neural and Statistical Classification,” Ellis Horwood Series in Artificial Intelligence, England, 1994.

[20] T. M. Mitchell, “Machine Learning,” McGraw Hill Series in Computer Science, USA, 1997.

[21] Y. Chen, D. S. Oliver, D. Zhang, “Data assimilation for nonlinear problems by ensemble Kalman filter with reparameterization,” Journal of Petroleum Science and Engineering, vol. 66, pp. 1-14, 2009.

[22] I. H. Witten and E. Frank, “Practical Machine Learning Tools and Techniques,” Morgan Kaufmann, San Francisco, 2005.

[23] H. S. Roh, J. O. Kim, “Manipulator Modeling from D-H Parameters,” 30th Annual Conference of IEEE Industrial Electronics Society, Busan, Korea.