Bayesian Network Classifier with Efficient Statistical Time-Series Features for the Classification of Robot Execution Failures

José Alonso-Tovar[#1], Baidya Nath Saha[*2], Jesús Romero-Hdz. [#3], David Ortega[*4]

[1,3,4]Centro de Ingeniería y Desarrollo Industrial (CIDESI), México.

[2]Centro de Investigación en Matemáticas (CIMAT), México.

Abstract. *— Accurate classification of robot execution failure during the manufacturing assembly operations guides to automate robot to perform the predefined tasks. In this paper we exploit the statistical transformations of time-series data for the classification of robot execution failure in the context of peg-hole insertion task. The statistical transformation of time-series data aims to reduce the dimension and unearth the discriminative features for the classification task and and hence improves the performance, such as predictive accuracy and Learning time. We collected force-torque sensor data for different execution failures during peg-hole insertion task using the industrial high speed and powerful MOTOMAN-MH6 six axis robot. We conducted an extensive supervised classification analysis with different classifiers with raw force torque sensor data as well as statistical features computed from the force torque sensor data. Experimental results demonstrated that Bayesian network classifier with efficient time-series features can more accurately classify different robot execution failures than other classifiers. We validated the experimental results on UCI benchmark dataset.*

Keywords — *Time series classification, Robot execution failures, Data transformation.*

I. Introduction

Parts mating, peg-hole-insertion or assembly operation is the most common operation in industry production, but autonomous execution by robots can significantly increase overall productivity. In manufacturing assembly operations, robots must successfully plan and execute tasks in the presence of uncertainty, for example positioning uncertainty. Since parts mating or peg-hole-insertion tasks involve object interactions, these tasks are subject to uncertainty arising from imperfect sensing and effecting such as object-position sensing and control, incomplete and faulty world models, and exogenous events [5]. There are several aspects of manufacturing assembly tasks that make the automation difficult including the requirement of high sensor capability, the complexity of the assembly task programming, even for the execution of the simplest tasks, at the cost of the use of expensive hardware, sensors and the application of advanced control techniques, and the limited autonomy and flexibility of industrial robots, that results in performance and success rates much below those of humans [3]. Human may achieve this task with much less time and fewer trials. It will be a great benefit if robots can learn the human skill and apply it autonomously [28].

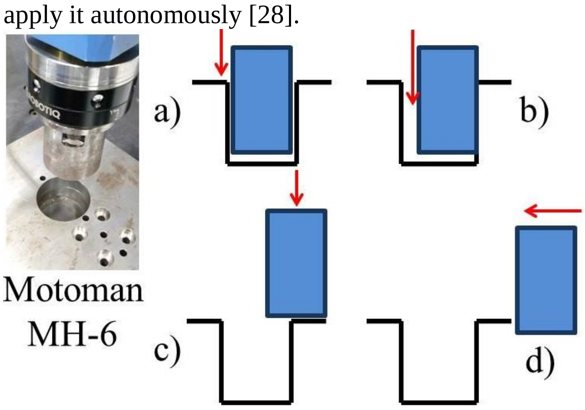

Fig. 1 (a) Normal, (b) jamming, (c) obstruction and (d) front collision.

Similar to humans, learning is prerequisite for cognitive robots to gain experience, to adapt to the real world and skills are acquired and improved through learning [17]. A cognitive robot owns abilities to plan to attain its goals, to execute its plan and to reason about dynamic cases. Due to unexpected outcomes plan execution may fail in the physical world. Robustness is central for success, and the robot should use the experience gained from the physical world in its future tasks [18]. To gain insight in problems of industrial peg-hole insertion operations, an industrial MOTOMAN-MH6 robot with six degrees of freedom was used to insert a cylindrical peg-hole insertion tasks. We identified three different execution failure situations (jamming, obstruction and front collision) during peg-hole insertion operations due to non-deterministic actions or different sources of uncertainty in physical dynamic environments as shown in Fig. 1. A monitoring system is essential to achieve goals robustly in the face of uncertainties. In the peg-hole insertion task, force-torque sensor located at the gripper generates abnormal force-torque signals during execution failure situations. Along with the detection of robot execution failures, the robot should also have the ability to detect the reasons of the execution failure for efficient recovery [7]. An inference process is required to segregate the failure to identify the underlying reason behind the failure. Automatic classification of robot execution failures using the force-torque data is the primary step of robot learning which can teach the robot to automatically identify the reasons or types of execution failures, modify the plan accordingly to avoid such situations and also guides the robot in the process of failure recovery (to take the suitable corrective actions immediately to get rid of the execution failures during manufacturing assembly operations such as peg-hole insertion task).

The reliability of a robot depends on the interactive ability between robot and the changing environmental conditions. The prediction ability of robot execution failures is imperative in the robotic field in the complex environments in which execution failures can have devastating consequences for robots and the objects in the surroundings. However, the prediction of robot execution failures is a difficult learning task for two reasons: (a) the partially corrupted or incomplete measurements of data and (ii) some prediction techniques are not suitable for predicting the robot execution failures with little samples [21].

In this research we demonstrated that Bayesian network classifier can accurately classify the robot execution failures with partially corrupted data [26][2]. Success of Bayesian network classifier in a wide range of applications such as gene expression [16], remote sensing imaging [27] and distance education system [8] has attracted to develop a Bayesian network classifier in the robot execution failures. However, learning the Bayesian network classifier in the high dimensional feature space such as force torque sensor data in the peg-hole insertion task is computationally very expensive. Hence, we exploited the statistical transformations of the force torque data (mean, sigma, skew, and kurtosis of the force and torque as well as the differential of the force and torque) for the classification of robot execution failure in the context of peg-hole insertion task. This statistical feature reduces the dimensionality of the force torque data and thus facilitates the implementation of the Bayesian network in peg-hole insertion task. Dimensionality reduction of the data expedites both the structure and parameters (the values of the conditional probability tables) learning of the Bayesian network structure. In addition, the statistical time-series features (force torque signal can be considered as time-series data since force and torque signal received by the robot gripper are recorded at regular intervals) significantly improve the quality of the data by reducing the effects of noise through averaging out of the noise. These statistical time-series features are more discriminative in nature that enhances the performance of the classifiers. We conducted and collected force-torque sensor data for three different execution failures (jamming, obstruction and front collision) and during normal execution for peg-hole insertion task using the industrial MOTOMAN-MH6 robot with six degrees of freedom [24]. We also conducted the experiment on the benchmark robot execution failure datasets from UCI machine learning repository [23]. We conducted a sensitivity analysis with eight different classifiers including Support Vector Machine (SVM), neural network or multilayer perceptron, decision tree, bagging, boosting, KStar, Tree Augmented Naïve Bayes (TAN) and Naïve Bayes (most simple form of Bayesian network). Experimental results demonstrate that TAN and Naïve Bayes with statistical features outperform the other classifiers with both statistical raw sensor (force-torque feature) data and statistical time-series data. TAN and naïve Bayes classifier with statistical features outperforms other classifiers on both Motoman MH-6 and UCI benchmark dataset.

The outline of the remaining of the paper is as follows. Section II reveals the literature review. Section III presents the Bayesian Network classifier and different learning algorithms. Section IV illustrates the statistical transformation of the force-torque sensor data. Section V demonstrates the experimental results and discussions. Section VI concludes this work.

II. Literature Review

Different machine learning based robot execution failures prediction models from incomplete and erroneous sensor data available in the literature are presented below.

Li *et al.* [21] proposed a novel Kernel Based Extreme Learning Machine (KELM) algorithm for robot execution failures. They formulated the robot execution failures as partially corrupted or incomplete measurements of data and utilized particle swarm optimization approach to optimize the parameters of kernel functions of neural networks for robot execution failures classification.

Karapinar *et al.* [17] developed an Inductive Logic Programming (ILP) learning based robot execution failures paradigm to frame hypotheses represented in first-order logic that are useful for further reasoning and planning processes. Background knowledge and partially specified world states was represented by these hypotheses which made this a generalized paradigm.

Koohi *et al.* [19] analyzed the impact of erroneous data for predicting the classification of robot execution failures. They have demonstrated that classification prediction accuracy increases with meta-level classifiers (Boosting, Bagging, Plurality voting, stacking using Ordinary Decision Tree (ODT) and stacking using Meta Decision Tree (MDT)) in noisy robot execution failure data associated with a humanoid robot.

Beetz *et al*. [4] developed a robot execution planner which diagnoses projected plan failures by classifying them in taxonomy or predefined failure models. This plan avoids the consequences of unusual but predictable adverse circumstances by predicting and transforming the plan that makes the plan more robust in a changing and partially unknown environment.

Altan *et al*. [2] investigated the root cause of robot execution failure detection and proposed a temporary fault isolation based hierarchical hidden Markov model (HHMM) was proposed. This HHMM was run in parallel to determine the causes of unexpected deviations.

Twala *et al*. [29] proposed a probabilistic classification approach for the classification of incomplete, partially corrupted and inconsistent robot execution failure data. By improving the estimated probabilities, our approach offers substantial computational savings and increases the classification performance.

Diryag *et al*. [11] developed the prediction of robot execution failures model based on neural networks. Forces and torques recorded immediately after the system failure is used and multilayer feedforward structure are trained for neural network training.

III. BAYESIAN NETWORK CLASSIFIER FOR ROBOT EXECUTION FAILURES

Let $U = \{F_{x_1}, F_{y_1}, \ldots, T_{z_k}\}$, be a vector of force-torque signal at *k* different time point. A *Bayesian network B* over a set of variables *U* is a network structure B_S, which is a directed acyclic graph (DAG) over *U* and a set of probability tables $B_P = \{p(u|pa(u)) | u \in U\}$ where $pa(u)$ is the set of parents of u in B_S. A Bayesian network represent a probability distributions $P(U) = \prod_{u \in U} p(u|pa(u))$.

The classification task consist of classifying a variable $y = y_0$ called the class variable given a set of variables $f = f_{x_1} f_{y_1} \ldots t_{z_k}$ called attribute variables. In this robot failure execution task, f represents the force and torque values sensed by the gripper along three different axes, *x*, *y*, and *z* at *k* different time intervals and y takes any of the values of normal, collision, jamming, front collision or obstruction. A classifier $h : \rightarrow y$ is a function that maps an instance of *f* to a value of y. The classifier is learned from a dataset D consisting of samples over (f, y). The learning task consists of finding an appropriate Bayesian network given a data set D over U.

Inference algorithm for robot execution failures

To use a Bayesian network as a classifier for robot execution failures, we need to calculate $\arg\max_y P(y|f)$ using the distribution $P(U)$ represented by the Bayesian network. Here it is noted that $P(y|f) = P(U) / P(f) \alpha\ P(U) = \prod_{u \in U} p(u|pa(u))$,

where all variables in *f* are known. We need to calculate $P(y|f)$ for all values of y.

Learning Bayesian network algorithms for robot execution failures

Bayesian network learning performs in two stages: first learn a network structure, then learn the Conditional Probability Tables (CPT).

A. Structure learning of the Bayesian network for robot execution failures

Various approaches to structure learning are available in the literature, we name the following four here.

Local score metrics: Learning a network structure B_S can be considered an optimization problem where a quality measure of a network structure such as Bayesian approach, minimum description length, information and other criteria given the training data $Q(B_S|D)$ needs to be maximized. The score of the whole network can be decomposed as the sum (or product) of the score of the individual nodes and thus lead these metrics computationally tractable and also this allows for local scoring and local search methods as well [1].

Conditional independence tests: This method assumes that there is a network structure that exactly represents the independencies in the distribution that generated the data. There is no arrow between those two variables if any (conditional) independency can be identified in the data. These methods attempt to uncover the causal structure in the data. Conditional independencies in the data are properly represented by the direction of the edges [10].

Global score metrics: The performance of a classifier is measured on a given data set by predicting its future performance through the estimation of the expected utilities, such as classification accuracy. Cross validation method facilitates this out of sample evaluation strategy by repeatedly dividing the data in training and validation sets. Evaluation of a Bayesian network structure is conducted by estimating the network's parameters from the training set and evaluating the performance of Bayesian network against the validation set. Global scoring metric is computed by the average performance of the Bayesian network over the validation sets that provides the quality of the network. Computation of global scoring metrics for big network structure is sometimes computationally intractable because unlike local scoring metrics, cross validation often cannot be decomposed in the scores of the individual nodes.

Hence, the complete network structure needs to be considered while computing the global metric [22].

Fixed structure: The simplest learning strategy is to consider a fixed structure and the structure can be fixed in many applications through the integration of domain knowledge. Naïve Bayes belongs to a fixed structure.

Different search algorithms such as hill climbing, simulated annealing and tabu search [6] are used in all of this structure learning process. Once a good network structure is learned, the next task is to estimate the conditional probability tables for each of the variables.

B. Learning Conditional Probability Table (CPT) for Bayesian network structure for robot execution failures

We describe here two different approaches of learning CPT.

Simple Estimator: The Simple Estimator [15] class produces direct estimates of the conditional probabilities,

$$P(f_i = k | pa(f_i) = j) = \frac{N_{ijk} + N'_{ijk}}{N_{ij} + N'_{ij}}$$

where N'_{ijk} is the alpha parameter which is set as 0.5 (by default). Maximum likelihood estimates is obtained with alpha=0.

Bayes Model Averaging: Bayesian Model Averaging (BMA) estimator estimates the conditional probability tables based on the Bayes model averaging of all network structures that are substructures of the network structure learned [9]. Here we estimate the conditional probability table of a node x_i given its parents $pa(x_i)$ as a weighted average of all conditional probability tables of x_i given subsets of $pa(x_i)$. The weight of a distribution $P(x_i|S)$ with $S \subseteq pa(x_i)$ used is proportional to the contribution of network structure $\forall_{y \in S} y \rightarrow x_i$.

Since the structure learning is computationally intractable for high dimensional datasets like raw sensor (force torque) dataset used in this experiment for robot execution failures we implement two simplified model of Bayesian Network classification: Naïve Bayes and Tree Augmented Naïve Bayes (TAN) in our research. The details of these two Bayesian network model is discussed below.

a) Naïve Bayes

A Naïve-Bayes Bayesian Network [12], is a simple structure that has the classification node as the parent node of all other nodes as shown in Fig. 2.

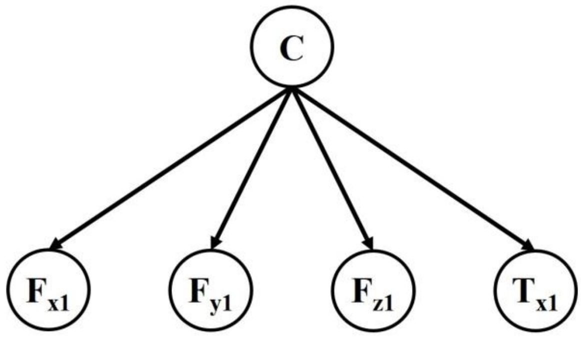

Fig. 2 Naïve Bayes structure

Though it is the most simplified model of Bayesian Network Classifier, Naive-Bayes has been used as an effective classifier over many years in a wide range of applications. Naïve Bayes possesses two major advantages. First, it is easy to construct, as the structure is given a priori and hence no structure learning procedure is required. Second, the classification process is very efficient. Both advantages are a result of its assumption that all the features are independent of each other given its class label. Although this independence assumption is not valid for many applications, however Naïve-Bayes outperforms many sophisticated classifiers in many applications, especially where the features are not strongly correlated [20].

The procedure of learning Naive-Bayes as shown in Fig. 1 is as follows:

1. Let the classification node be the parent of all other nodes.
2. Learn the parameters (the empirical frequency estimates) and output the Naïve-Bayes Bayesian Network.

b) Tree Augmented Naïve Bayes (TAN)

Let $U = \{F_{x_1}, F_{y_1} ...., T_{z_k}, c\}$, represent the node set (where c is the classification node) of the data. TAN classifiers [13] first learns a tree structure over $U \setminus \{c\}$, using mutual information tests conditioned on c. It then adds a link from the classification node to each feature node, similar to a Naïve-Bayes structure (i.e., the classification node is a parent of all other nodes) as demonstrated in Fig. 3. (Note that features $F_{x_1}, F_{y_1}, F_{z_1}, T_{x_1}$ form a tree.)

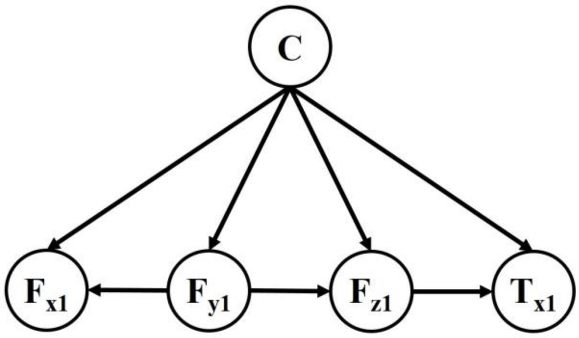

Fig. 3 The classification node is a parent of all other nodes

The procedure for TAN structure learning as demonstrated in Fig. 2 is as follows.

1. Take the training set and $U \setminus \{c\}$ as input.

2. Use the modified Chow-Liu algorithm to compute the conditional mutual information test. The original algorithm is modified by replacing every mutual information test $I\left(F_{x_i}, F_{x_j}\right)$ with a conditional mutual information test $I\left(F_{x_i}, F_{x_j} \mid \{c\}\right)$.

3. Add *c* as a parent of every F_{x_i}, where $1 \le i \le k$.

4. Learn the parameters and output the TAN.

This complete algorithm, which extends the Chow-Liu algorithm, requires $O\left(k^2\right)$ conditional mutual information tests.

IV. Statistical Transformation of the Force Torque Sensor Data

Force-torque sensor data can be considered as the time-series data that contains force and torque measurements collected over a period of time during the execution failure of the robot. We also exploit the statistical features of the time series data such as, mean (μ), variance (σ), skewness (γ) and kurtosis (κ) for classification of robot execution failures which are as follows [25]:

$$\mu = \frac{\sum_{t}^{n} y(t)}{n}, \sigma = \sqrt{\frac{\sum_{t=1}^{n} (y(t)-\mu)^2}{n}}$$

$$\gamma = \frac{\sum_{t=1}^{n} (y(t)-\mu)^3}{n\sigma^3}, \kappa = \frac{\sum_{t=1}^{n} (y(t)-\mu)^4}{n\sigma^4} - 3$$

Where, $y(t)$ is the force or torque measurement at time point *t*, μ, σ, γ, and κ represent mean, sigma, skew and kurtosis computed for each sample (force-torque signal) of a particular failure or normal execution. We also compute the μ, σ, γ, and κ for $y'(t)$ ($y'(t+1) - y'(t)$) as well.

V. Experimental Results

A. Data collection

1. Motoman MH-6 dataset

We conducted the experiment on industrial high speed, powerful, compact and efficient MOTOMAN MH-6 robot (payload 6 kg, repeat ± 0.08 mm) [24]. We executed three different failure executions (jamming, obstruction and front collision as well as normal condition as shown in Fig. 4.

(a) (b) (c) (d)

Fig. 4 (a) Normal, (b) jamming, (c) obstruction and (d) front collision.

Typical behavior of the force and torque signal along *x*, *y* and *z* axis are demonstrated in Fig. 5. From fig. 5 it is prevalent that front collision occurs along +y axis of the sensor. Hence, high values of $-F_y$ force and $+T_x$ torque are perceived by the sensor. Similarly, Obstruction occurs along +*z* axis and the offset occurs along both +*x* and +*y* direction. Hence, high values of $-F_z$ and $+T_y$, medium values of $+F_x$, $+F_y$ and $+T_y$ and low values of $+T_y$ and $-T_z$ are perceived by the sensor. For normal, the values of the signals of all forces and torques are very low and this values are obtained due to the noise of the process.

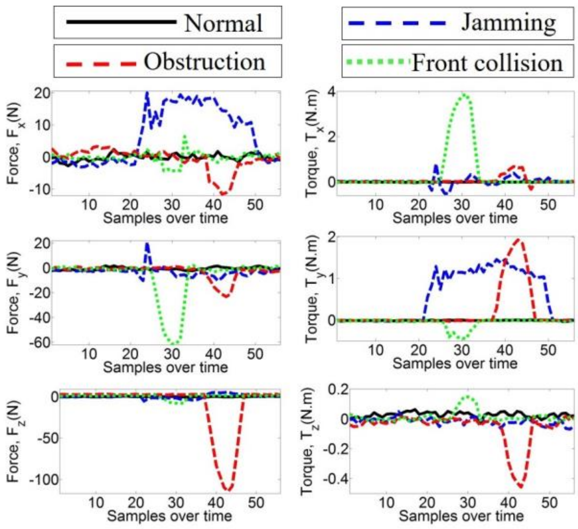

Fig. 5 Force and torque signals along x, y and z directions for normal, jamming, obstruction and front Collision for Motoman MH-6 dataset.

We collected 100 force torque signal samples (25 for each class: normal, jamming, obstruction and front collision) at a regular interval of 1 milliseconds with total 336 attributes. We collected force and torque in three different directions (along *x*, *y* and *z* axis) at each time point and we collected these force and torque measurements for 56 time points for each execution failure and normal execution.

2. UCI Robot Execution Failure Datasets

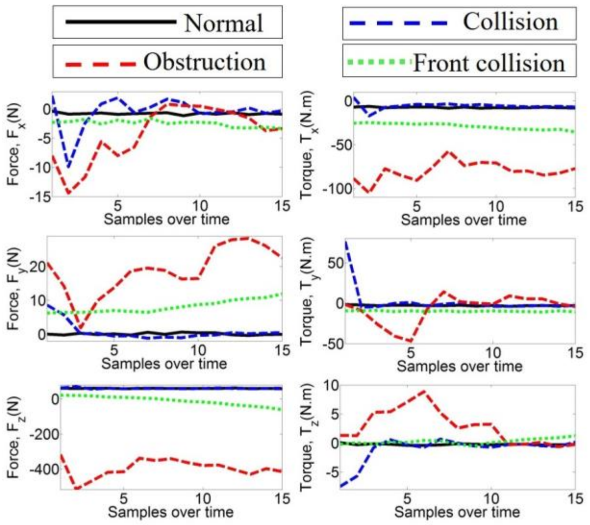

Fig. 6 Force and torque signals along x, y and z directions for normal, collision, obstruction and front Collision for UCI data.

This time-series dataset has been collected from UCI machine learning repository [23] that contains force and torque measurements on a robot after failure detection, 88 failures in approach to grasp position. Each failure instance is characterized in terms of 15 force/torque samples collected at regular time intervals starting immediately after failure detection with total 90 attributes (six attributes, three force and three torque measurements along *x*, *y* and *z* axis at each of 15 time points). The total observation window for each failure instance was of 315 ms. Data are distributed in four classes: normal, collision, front collision, obstruction as shown in Fig. 6.

B. Feature Vector Calculation

1. Force-torque feature vector

The format of each sample regarding each execution failure available in the UCI machine learning repository is as follows.

Class

$$\begin{bmatrix} Fx1 & Fy1 & Fz1 & Tx1 & Ty1 & Tz1 \\ \dots & & & & & \\ Fx15 & Fy15 & Fz15 & Tx15 & Ty15 & Tz15 \end{bmatrix}$$

First, an algorithm for the arrangement of raw sensor matrices (force torque data) was developed to obtain the attributes of each sample and merge them into a single vector, converged in 88 samples with 90 attributes.

2. Statistical feature vector

We compute four statistical time-series feature vectors: mean, sigma, skew and kurtosis for each execution failure and normal execution as discussed in section IV. We compute these four feature vectors for three force and three torque components as well as second-order features, *i.e.*, the differentials of all three force and three torque components which converged to 48 feature vectors for each execution sample. Whereas, the raw sensor data (force-torque measurements) consists of 336 attributes for Motoman MH-6 dataset and 90 attributes for UCI machine learning dataset, statistical time-series feature dataset consists of only 48 attributes and thus statistical transformation of the data significantly reduces the high dimensional complexity of the force-torque data. This dimensionality reduction facilitates the implementation of Bayesian network classifier for classifying the robot execution failures which is not practically feasible with high dimensional force-torque attributes.

In addition this statistical transformation of the force torque data improves the quality of the data by demonstrating better discriminative characteristics for robot execution failures.

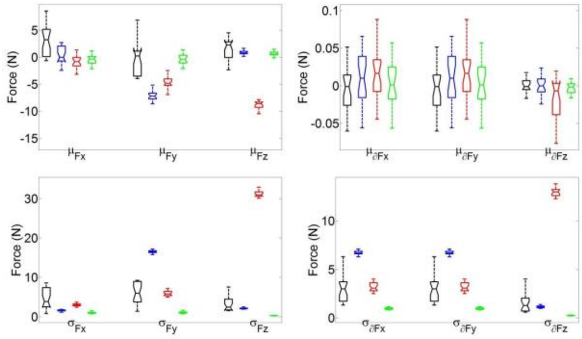

Fig. 7 Mean (μ) and Variance (σ) of the force (F) and its differentials (F´) for Motoman MH-6 dataset.

Fig. 7-10 shows the box plot of the time-series features of Motoman MH-6 dataset for four different classes (normal, jamming, obstruction and front collision) and Fig. 11-14 shows the box plot of the time-series features of UCI dataset for four different classes (normal, collision, obstruction and front collision). Different spread of different boxplots for different classes shows the discriminative power of the statistical transformation of the force-torque data. To demonstrate a few, the spread of the boxplots representing jamming (in blue color) in Fig. 7 and 8 are different from the spread of the box plots representing other classes. Similarly, Fig. 11 and 14 demonstrate that mean (μ) and kurtosis (κ) of the torque illustrates significant discriminative classification capability for obstruction (in red color) and collision (in blue color) respectively.

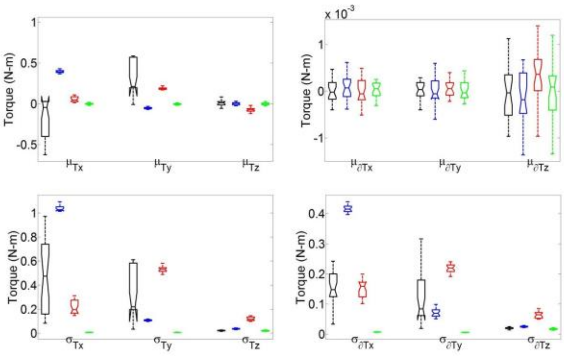

Fig. 8 Mean (μ) and Variance (σ) of the torque (T) and its differentials (T´) for Motoman MH-6 dataset.

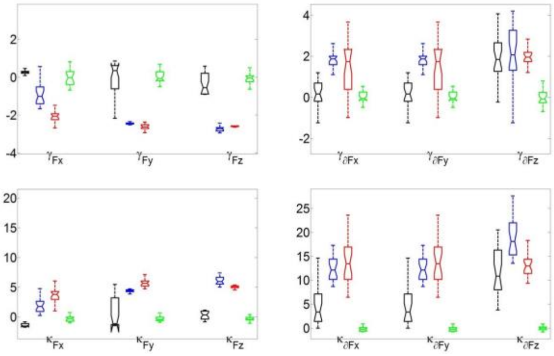

Fig. 9 Skew (γ) and Kurtosis (κ) of the force (F) and its differentials (F´) for Motoman MH-6 dataset.

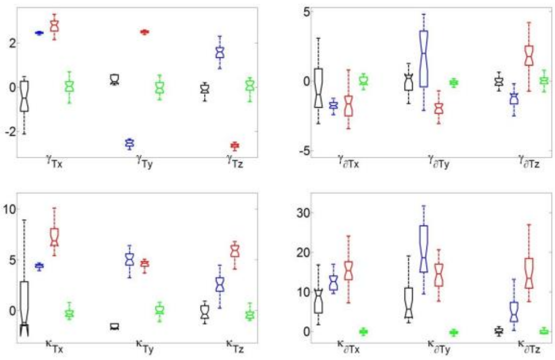

Fig. 10 Skew (γ) and Kurtosis (κ) of the torque (T) and its differentials (T´) for Motoman MH-6 dataset.

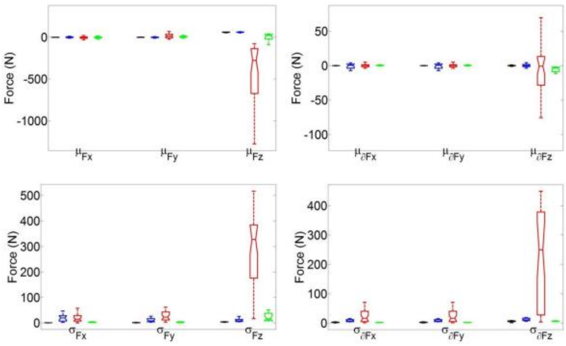

Fig. 11 Mean (μ) and Variance (σ) of the force (F) and its differentials (F´) for UCI dataset.

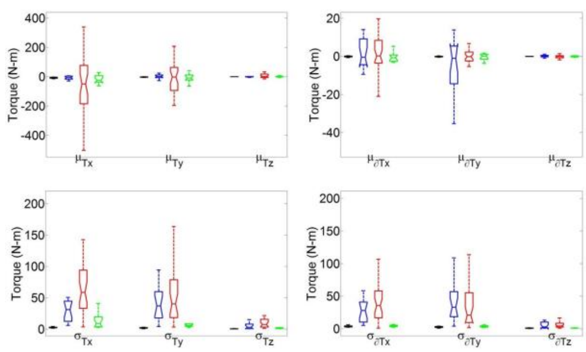

Fig. 12 Mean (μ) and Variance (σ) of the torque (T) and its differentials (T´) for UCI dataset.

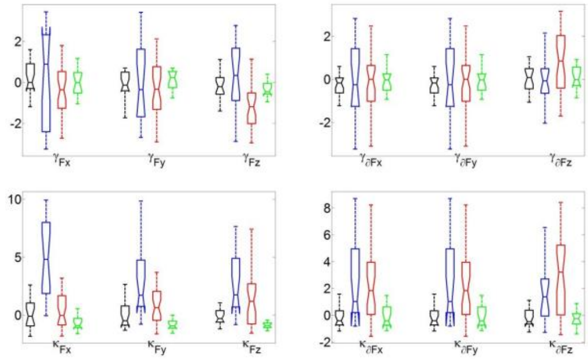

Fig. 13 Skew (γ) and Kurtosis (κ) of the force (F) and its differentials (F´) for UCI dataset.

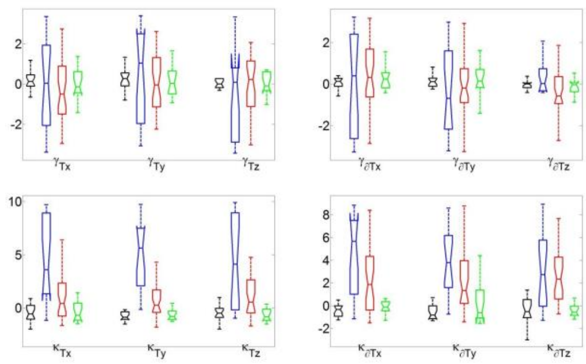

Fig. 14 Skew (γ) and Kurtosis (κ) of the torque (T) and its differentials (T´) for UCI dataset.

C. Sensitivity analysis of different classifiers

We conduct the sensitivity analysis of different machine learning based classifiers such as, neural network or multilayer perceptron, support vector machine (SVM), KStar, Bagging, AdaBoost M1, Decision Tree, Naïve Bayes and Tree augmented Naïve Bayes (TAN) for both Motoman MH-6 and UCI datasets. We use WEKA implementation for computing the performance of all classifiers. Classification performance of these classifiers in terms of Accuracy, F-Measure and ROC area demonstrated in Fig. 15-18 of both raw sensor (force-torque) and time-series features for both Motoman MH-6 dataset and UCI machine learning data. Results of these classification performance demonstrate that: (*i*) Bayesian network classifier (Naïve bayes and Tree

Augmented Naïve Bayes (TAN)) can classify more accurately different robot execution failures than other classifiers; (*ii*) Statistical transformation of time series data demonstrates the discriminative capability of classifying the robot execution failures and thus enhances the classification performance of all the classifiers as shown in Fig. 17 and 18; (*iii*) The Bayes TAN model using force torque features for both Motoman MH-6 and UCI datasets is taking unrealistic execution time for learning Bayesian TAN structure using personal computer (4 GB RAM, intel(R) core(TM) i3-4005U CPU @ 1.70GHz Dell laptop) and weka implementation [14] hence we could not include the performance of TAN using force torque data (90 and 336 attributes for UCI and Motoman MH-6 dataset) respectively. Hence the dimensionality reduction through statistical transformation of the force-torque features facilitates the implementation of TAN classifier in real time.

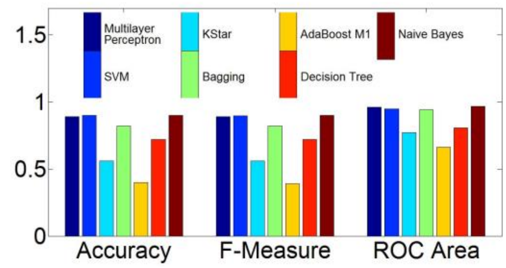

Fig. 15 Classification performance using force torque feature of Motoman MH-6 dataset.

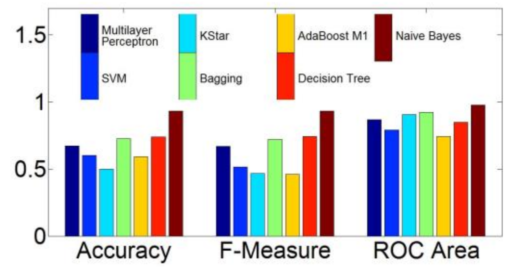

Fig. 16 Classification performance using force torque feature of UCI dataset.

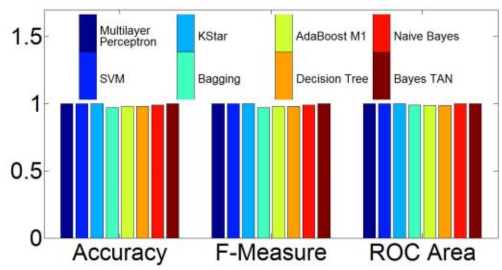

Fig. 17 Classification performance using time-series feature of Motoman MH-6 dataset.

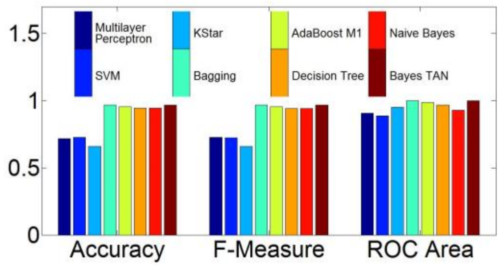

Fig. 18 Classification performance using time-series feature of UCI dataset.

Fig. 19 and 20 illustrate the decision tree model of Motoman MH-6 and UC dataset for both force-torque and time-series features. Fig. 21 demonstrates the Tree Augmented Naïve bayes (TAN) model for Motoman MH-6 dataset. Both decision tree and TAN model demonstrate that force and skew of the force lie in the top level of the tree and thus appear as the most discriminative features for classifying the robot execution failures which consense with the outcomes of the box plot illustrated in Fig. 7 -14.

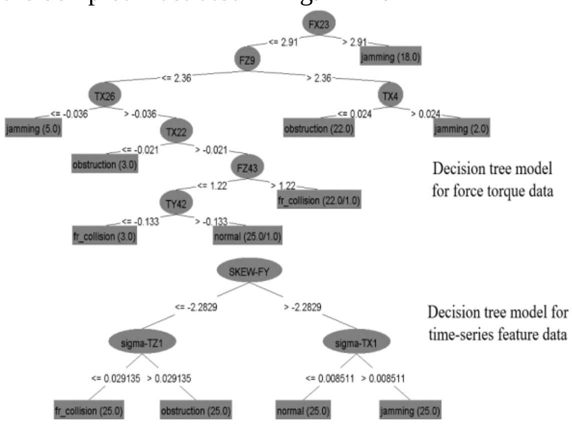

Fig. 19 Decision Tree model for Motoman MH-6 dataset.

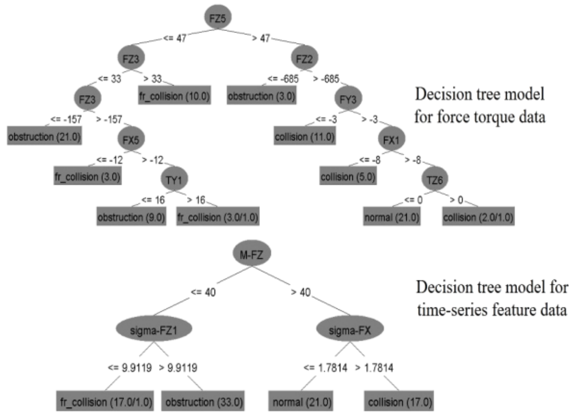

Fig. 20 Decision Tree model for UCI dataset.

VI. CONCLUSIONS

We used the statistical features (mean, sigma, skew, and kurtosis) of the force torque sensor data for the classification of robot execution failure in the context of peg-hole insertion task. This statistical feature reduces the dimension of the data significantly, improves the quality of the data by increasing the discriminative power that facilitates the classification task. Dimensionality reduction of the data helps to execute the learning of the Bayesian network structure and parameters (the values of the conditional probability tables) in real time. We conducted and collected force-torque sensor data for three different execution failures (jamming, obstruction and front collision) and normal execution during peg-hole insertion task using the industrial MOTOMAN-MH6 robot with six degrees of freedom. We also conducted the sensitivity analysis on benchmark UCI dataset. We have demonstrated that the TAN and Naïve Bayes (most simple form of Bayesian network) with statistical features outperforms the other classifiers including support vector machine, multilayer perceptron, decision tree, bagging and boosting with both statistical raw sensor (force torque) data. TAN and naïve Bayes classifier with statistical features outperforms other classifiers on UCI benchmark dataset. In future, we would like to implement fuzzy logic and auto-regressive learner to teach the robot taking corrective action automatically during execution failures while conducting manufacturing assembly operation.

ACKNOWLEDGMENT

We acknowledge Conacyt for supporting our research.

REFERENCES

[1] M. Abramovici, M. Neubach, M. Fathi and A. Holland, "Competing Fusion for Bayesian Applications", *Proc. IPMU*, pp. 378–385, 2008.

[2] D. Altan and S. Sariel-talay, "Probabilistic Failure Isolation for Cognitive Robots," pp. 370–375, 2009.

[3] L. Balletti, A. Rocchi, F. Belo, M. Catalano, M. Garabini, G. Grioli, and A. Bicchi, "Towards variable impedance assembly : the VSA peg-in-hole." , IEEE-RAS International Conference on Humanoid Robots (Humanoid), pp. 504-508, 2012.

[4] M. Beetz and D. McDermott, "Improving Robot Plans During Their Execution.," *Aips*, pp. 7–12, 1994.

[5] J. F. Broenink and M. L. J. Tiernego, "Peg-in-Hole assembly using Impedance Control with a 6 DOF Robot," pp. 504–508, 1996.

[6] R. Bouckaert, "Bayesian network classifiers in Weka," *Dep. Comput. Sci. Univ.*, pp. 1–23, 2004.

[7] L. M. Camarinha-Matos, L. S. Lopes, and J. Barata, "Integration and learning in supervision of flexible assembly systems," *IEEE Trans. Robot. Autom.*, vol. 12, no. 2, pp. 202–219, 1996.

[8] M. Da, W. Wei, H. Hai-guang, and G. Jian-he, "The Application of Bayesian Classification Theories in Distance Education System," no. July, pp. 9–16, 2011.

[9] D. H. Dash, and G. F. Cooper, "Model Averaging for Prediction with Discrete Bayesian Networks", *J. Mach. Learn. Res.*, vol. 5, pp. 1177–1203, 2004.

[10] L. M. De. Campos, "A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests", *J. Mach. Learn. Res.*, vol. 7, pp. 2149–2187, 2006.

[11] A. Diryag, M. Mitic, and Z. Miljkovic, "Neural networks for prediction of robot failures," Journal of Mechanical Engineering Science, vol. 228, issue. 8, pp. 1444-1458, 2014.

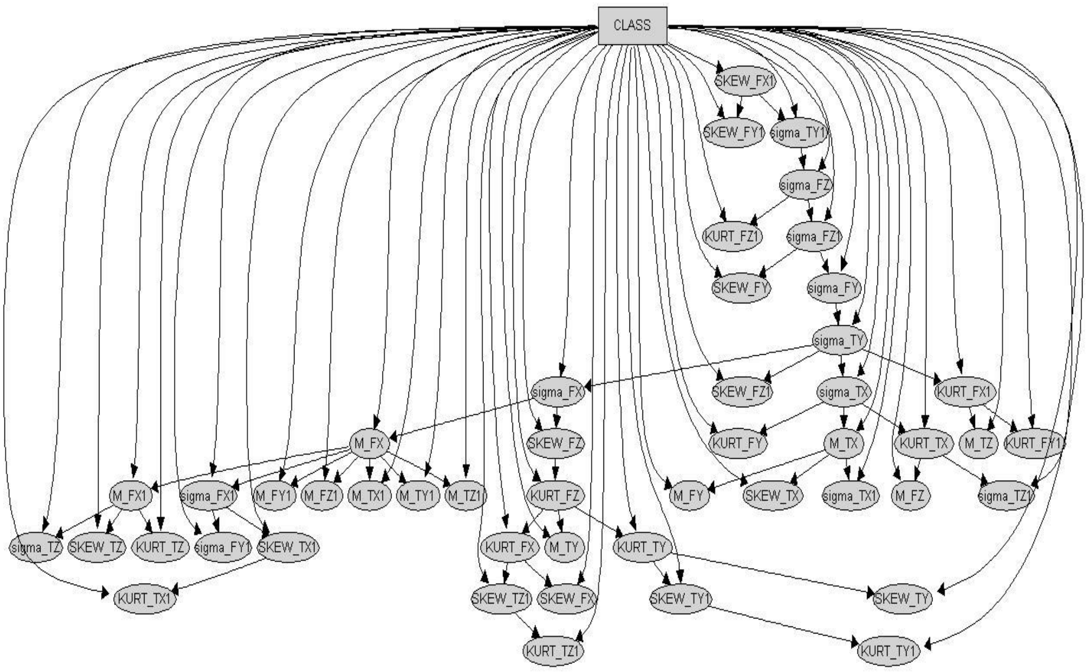

Fig. 21 Tree augmented Naïve Bayes (TAN) for Motoman MH-6 Robot Execution failure data.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," *New York: John Wiley, Section*. p. 680, 2001.

[13] N. Friedman, D. Geiger, and M. Goldszmit, "Bayesian Network Classifiers," *Mach. Learn.*, vol. 29, pp. 131–163, 1997.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an

update", *ACM SIGKDD explorations newsletter, 11*(1), 10-18, 2009.

[15] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data", *Mach. Learn. 20, 197-243*, vol. 243, pp. 197–243, 1995.

[16] P. Helman, R. Veroff, S. R. Atlas, and C. Willman, "A Bayesian network classification methodology for gene expression data," *J. Comput. Biol.*, vol. 11, no. 4, pp. 581–615, 2004.

[17] S. Karapinar and S. Sariel, "Cognitive robots learning failure contexts through real-world experimentation," *Auton. Robots*, vol. 39, no. 4, pp. 469–485, 2015.

[18] S. Karapinar, S. Sariel-talay, P. Yildiz, and M. Ersen, "Learning Guided Planning for Robust Task Execution in Cognitive Robotics," pp. 26–31, 2013.

[19] T. Koohi, E. Mirzaie, and G. Tadaion, "Failure Prediction Using Robot Execution Data," no. 1999, pp. 1–7, 2011.

[20] P. Langley, W. Iba, and K. Thompson, "An Analysis of Bayesian Classifiers," *AAAI-92 Proc.*, pp. 223 –228, 1992.

[21] B. Li, X. Rong, and Y. Li, "An improved kernel based extreme learning machine for robot execution failures," *Sci. World J.*, vol. 2014, 2014.

[22] Z. Liu, B. Malone, and C. Yuan, "Empirical evaluation of scoring functions for Bayesian network model selection", *BMC Bioinformatics*, vol. 13 Suppl 1, no. Suppl 15, p. S14, 2012.

[23] M. Lichman, "UCI Machine Learning Repository" [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[24] T. Messay, R. Ordóñez, and E. Marcil, "Computationally efficient and robust kinematic calibration methodologies and their application to industrial robots," *Robot. Comput. Integr. Manuf.*, vol. 37, pp. 33–48, 2016.

[25] A. Nanopoulos, R. Alcock and Y. Manolopoulos, "Feature-based classification of time-series data", *Inf. Process. Manag.*, vol. 0056, pp. 49–61, 2001.

[26] B. Qin, Y. Xia, S. Wang, and X. Du, "A novel Bayesian classification for uncertain data," *Knowledge-Based Syst.*, vol. 24, no. 8, pp. 1151–1158, 2011.

[27] C. Solares, A. N. A. Mar, and I. A. Sanz, "Bayesian Network Classifiers. An Application to Remote Sensing Image Classification," vol. 2005, pp. 62–67, 2005.

[28] T. Tang, H. Lin, and M. Tomizuka, "A learning-Based Framework for Robot Peg-Hole-Insertion," *ASME 2015 Dynamic Systems and Control Conference*, 2015.

[29] B. Twala, "Robot execution failure prediction using incomplete data," *2009 IEEE Int. Conf. Robot. Biomimetics, ROBIO 2009*, pp. 1518–1523, 2009.