2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
17-20 October, 2021. Melbourne, Australia

Building a Classifier Model for Failure Modes from Robot Sensor Readings through a Modified Forward Stepwise Algorithm

Brayden Kent, Maciej Łącki, Carlos Rossa

2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-4207-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMC52423.2021.9658748

***Abstract*—One of the many challenges in autonomous robots is that they can enter an error state and are unable to continue operation without human intervention. Sensors installed on the robot enable proprioception and could help the robot understand its error configuration. This paper proposes a method to determine from these sensor measurements, which are most critical in differentiating the error states such that the robot could understand its predicament, and could attempt at recovering without human aid. A classification model is built using the forward stepwise method and a scoring metric to overcome indecision in choosing between different features. This modified method is applied to three robot operating mode data sets. The experiments indicate an improvement to the classifier performance when using this the model built by the method compared to using all available predictor variables (features). With further refinement, this scoring metric could be a simple yet effective way to build classification models for increasing robot autonomy.**

***Index Terms*—Feature Extraction, Robot Failure Modes, Discriminant Analysis, Fault Detection and Diagnosis**

I. Introduction

Robotic automation increased efficiency, precision, and safety in industrial settings by replacing humans in dangerous, difficult, and repetitive tasks. Robots are gaining the ability to automate non-repetitive, complex, and abstract tasks such as controlling autonomous vehicles [1] or in autonomous robotic surgery [2]. A key aspect of advancing robotic technology where it can be widely implemented in these challenging applications is the development of robust fault detection and prevention. Robotic system robots are subject to actuation faults, such as singular configurations and actuator deterioration [3] or execution failures like insurmountable obstacles. When operating in an industrial setting such fault may cause costly equipment damage or manufacturing downtime, and with surgical applications, human life is on the line. It is, therefore, imperative for the robot to be able to identify its own faults and diagnose them.

In the field of Failure Detection and Diagnosis, there are three general classes of methods: knowledge-based, model-based, and data-driven approaches [4]–[6]. Knowledge-based systems use similar problem-solving methods as humans. For instance, applying causal analysis draws relationships between faults and symptoms such that when a certain predefined fault behavior is observed the system can easily identify it. This method successfully automates the fault detection task but it is difficult to implement as it requires a thorough understanding of a system and its faults. Once implemented, the system can only identify known faults and, to the authors' knowledge, has not been a method used for fault detection in robotic manipulators. Expert system-based methods, on the other hand, use a series of *if-then* statements. [7] present an expert-system-based framework for failure mode analysis, fault detection, tolerance, reconfiguration, and repair in a robotic manipulator. Due to the binary, *true* or *false*, terms used in their formulation, expert systems are sensitive to uncertainties [6]. One way to improve the robustness of the system is using a fuzzy expert system introduced in [8], where binary statements are replaced with graded ones.

Knowledge-based systems struggle to identify unknown faults as it requires supervised human expertise to determine what faults the robot expects, which makes them difficult to implement and scale to more complex systems, and consequently an unpopular method for fault detection and diagnosis in robotic systems.

The most popular, the model-based approaches, use a model of the system to simulate processes to detect and identify possible faults in operation, both known and unknown [6], [9]. For instance, parameter estimation identifies system parameters in the absence of faults. During system operation, the parameters are recalculated and a fault is identified if the value of the parameters changes. This method has been proposed in [10] for fault diagnosis of a 3-DOF industrial robotic manipulator. Observer-based methods, on the other hand, use system models to predict system behavior and compares it with the actual behavior of the system. A fault is identified when the two signals do not match. Such a method has been used for robotic

B. Kent, M.Łącki, and C. Rossa are with the Faculty of Engineering and Applied Science, Ontario Tech University, Oshawa, Ontario, Canada. E-mail: brayden.kent@ontariotechu.net; maciej.lacki@ontartiotechu.net; carlos.rossa@ontariotechu.ca.
We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Institutes of Health Research (CIHR), and the Social Sciences and Humanities Research Council of Canada (SSHRC), [funding reference number NFRFE-2018-01986]
Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), par les Instituts de recherche en santé du Canada (IRSC), et par le Conseil de recherches en sciences humaines du Canada (CRSH), [numéro de référence NFRFE-2018-01986]

manipulators. For instance, [11], [12] present the use of a nonlinear observer for fault detection in a robotic manipulator with unmodelled dynamics and discretization errors.

Model-based fault detection methods are primarily limited by the accuracy of the model. As the complexity of the system and the nonlinear behavior dominates the dynamics of a manipulator it becomes difficult to capture the true dynamics of the device. Even if a highly accurate device model exists it may be impractical to implement it due to its high computational cost. As such, model-based methods are best suited for moderately complex but well-understood systems. In complex systems, a different class of methods must be used such as the data-driven methods.

Data-driven approaches identify faults without reliance on a device model. Instead, they use sensors and device state data directly to identify known and unknown faults. One way of achieving this goal is by using neural networks. For instance, [13], [14] proposes the use of a Sigmoid Neural Network along with the device model to not only identify faults but also adjust the device model once the faults appear. Another approach has been presented in [15] where Self Organising Map Neural Network is used with a Radial Basis Neural Network to detect faults in robotic manipulators and predict its servicing requirement. A comprehensive analysis of neural network use in the field of robotic is presented in [16].

With an overwhelming amount of data available in a robotic system, it is crucial to minimize computation time and promote efficiency by prioritizing only data relevant to fault detection. As a result, dimensionality reduction methods are commonly used when training a classifier [17]. The reduced model is not only quicker, and more accurate, but also easier to interpret. The method proposed in [18] allows a designer to interpret the model features and their functions. On the other hand, a method proposed by [19] improves the robustness of the dimensionality reduction methods in systems with many dimensions and ample noise.

One method that has not been used, thus far, for fault identification is the stepwise model building method proposed in [20]. It is a good candidate for fault identification as it is simple to implement and accurate. However it can struggle, in some cases, to select between two features with equal chances of improving the model.

This paper presents a novel data-driven method for identifying failure modes in a robotic manipulator. The method is based on the forward stepwise model building method presented in [20] and it incorporates a novel advancement [21] scoring criteria that improve the model building by helping to choose between features that could improve the model. The proposed method refinement improves the accuracy of the model with the potential to use a fewer number of features than all of the available predictive variables. As a result, the features extracted by this method could still represent physical measurements making it easier to understand and adapt to many applications.

To the best of our knowledge, the forward stepwise model building method and its derivation have not been used for robot failure detection and identification, thus far. Therefore, this paper presents a first-of-its-kind implementation and evaluation, and a comparison of the forward stepwise model and the proposed modification.

To this end, Section II presents the modified forward stepwise model building method and its working principles. The method is then validated, in Section III by comparing its prediction capabilities with a model that uses all available features, in detecting failures in robotic manipulators using three data sets. Section IV then discusses the results, highlighting the performance difference between the two methods. Finally, Section V concludes the applicability and feasibility of using the novel method in various applications.

II. Classifier Model

A. Building the Classification Model

The classifier model is constructed using a method similar to the forward stepwise analysis method [20]. One begins with a null model; a model that contains no features. One at a time, each possible feature available in the training data set is added to the model, and the predictive accuracy of the model is tested with the validation data set. In the basic forward stepwise analysis method, the feature with the largest improvement to the predictive accuracy is permanently added to the model. The process then repeats, where additional features are added until some type of stopping criteria is achieved.

One possibility that can arise in this method is that multiple features may yield the same amount of improvement to the model, see Figure 5. One must then introduce a decision criteria to choose between which of the candidate features to add to the model. Here, a decision criteria originally introduced in [21] could be used. This selection method utilises the available metrics in the training data to determine which features are more likely to share similar values across the classes.

For a given feature x, the amount of which two classes c_1 and c_2 having similar values can be quantified with the overlapping index η see Figure 1. The overlapping index is evaluated by computing the area underneath the intersection of the probability density functions for these classes [22], in essence,

$$\eta(x|c_1,\, x|c_2) = \int_{-\infty}^{\infty} \min\big(f(x|c_1)\,,\; f(x|c_2)\big)\, dx. \quad (1)$$

Since both $f(x|c_1)$ and $f(x|c_2)$ represent probability density curves, the value of the overlapping index is $\eta \in \mathbb{R}[0,\, 1]$. An overlapping index of 0 would indicate that the feature has no commonality between the two classes. In contrast, a value of 1 would indicate that the two classes have exactly identical values for the feature. In the context of classification, it is desirable to have a low value for the overlapping index between two features such that each feature is unique. However, a low value for the overlapping index is an insufficient marker on its own for extracting unique features where η could be a small value caused by a large range in one of the classes.

Adding the coefficient of variation v as a metric introduces a penalty for data that have high variance. This can be calculated using the mean value μ and standard deviation σ for i^{th} feature in the j^{th} class such that,

$$v(x_i|c_j) = \left|\frac{\sigma}{\mu}\right|. \tag{2}$$

A smaller value of v is desirable, as it would indicate that the data is more closely gathered about the mean, and would suggest that future measurements would also be near the mean value.

With the overlapping index and coefficient of variation metrics defined, the scoring metric proposed in [21] can be used to quantify the similarity the values for a i^{th} feature of two arbitrary classes would be,

$$SC(x_i|c_1,\, x_i|c_2) = \eta(x_i|c_1,\, x_i|c_2)v(x_i|c_1)v(x_i|c_2) \tag{3}$$

B. Classifier Method Selection

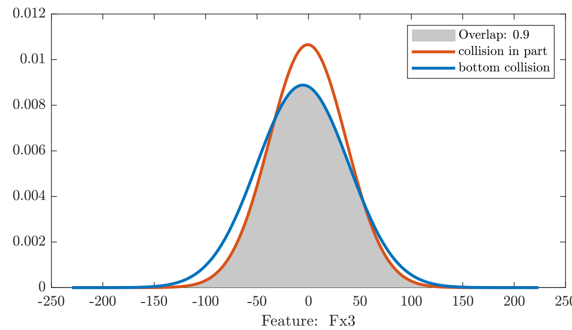
There are many valid classifiers that exist in the literature. In this paper, a discriminant analysis classifier is used. This type of classifier is easy to evaluate, and is applicable to multi-class problems. At its core, this classifier establishes boundary curves that separate the classes, and determines the label for a new test sample by determining on what side of the boundary the new test sample lies. These boundary curves can be constructed simply by knowing the mean and covariance of the training data points for each class. Since it is not a guarantee that the covariance for each class will be equal, linear boundary curves can not be used. In this circumstance, quadratic boundary curves can be used instead.

III. Method Validation

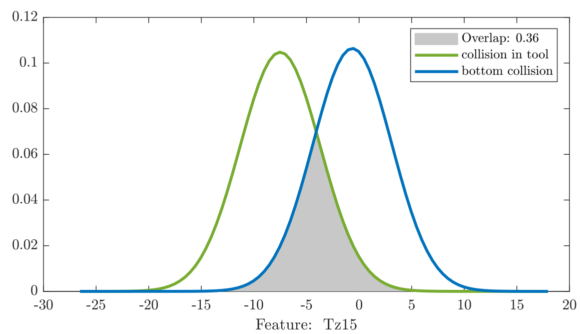
A. Robot Failure Mode Database

The database provided by Luis Seabra Lopes and Luis M. Camarinha-Matos [23] provides 6-DOF measurements from 15 sensors on a robot, totalling 90 features and an associated robot status label for a robotic manipulator during its programmed tasks. Using this data set, one could conceivably develop a classification model using these features to predict the robot's status given a new set of measurements. However, using all of the available measurements may not be desirable. A better classification model could be developed with fewer features: using only features that best separate the classes.

This paper utilises the LP1, LP4 and LP5 data sets from this repository; LP2 and LP3 were excluded from this study as there were an insufficient number of instances for all classes to properly train the classifier. The LP1 data set contains four unique classes that denote the state of the robot. Similarly, LP4 contains three classes, and LP5 contains five classes.



(a)



(b)

Fig. 1: (a) The probability density functions for two classes from the training data of the LP5 data set, featuring a large amount of overlap. This indicates that these classes will likely share similar values for the feature. (b) From the same training data set, two classes with relatively smaller overlap. This indicates, at this feature, the classes are less likely to share similar values.

B. Applying the Method

The forward stepwise analysis method with the selection criteria described above was used to construct a model for a quadratic discriminant analysis classifier. The stopping criteria used in this study are: 1) the addition of an additional feature would make the predictive accuracy of the model worse, 2) the number of features in the model being built exceeds the arbitrarily selected maximum of eight features.

C. Experimental Results

The three data sets investigated were each analysed nine times. For every run, the data was randomly partitioned into test and training data with a 70-30% split respectively. The training data was further partitioned with a validation test set, also with a 70-30% hold out.

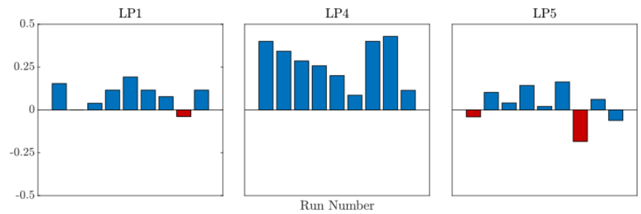


Fig. 2: A graph where the vertical axis depicts the difference in model accuracy using all features and the accuracy from the extracted features. The general trend seen across 9 runs with each data set suggests an improvement to the classifier accuracy when using the extracted features. The LP4 data set in particular was highly receptive to the method, where the model was consistently improved by a significant margin. The LP1 and LP5 data sets had experienced a few instances where the classifier accuracy was worse when only using the extracted features.

Each run would utilise the method described in this paper to build the classifier model with the extracted features. Shown in Figure 2 is the difference in model accuracy when using all of the features in the data set and only the extracted features. The average accuracy with built models for the LP1, LP4 and LP5 data sets were 79%, 81%, and 59% respectively. For comparison, the average accuracy of the models that used all features for the LP1, LP4 and LP5 data sets were 71%, 53%, and 56%.

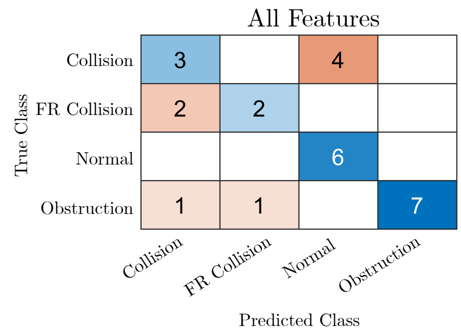
The method was able to reduce the number of features in the classifier in all of the simulation runs due to the applied stopping criteria.

Shown in Figure 3 is the result of one of runs with the LP1 data set. The confusion matrices highlight an improvement in classifier accuracy when using the model with extracted features. Similar performance was seen in other runs, where the number of misclassifications was reduced when using the model with extracted features.

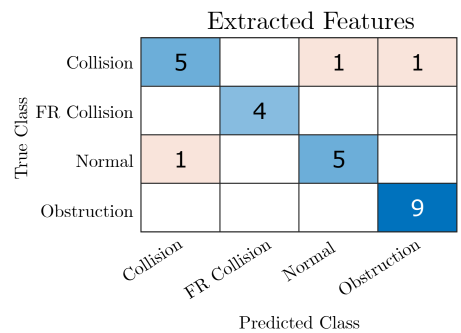
IV. Discussion

The proposed method used in this paper appear to be useful at reducing the feature space of the robot failure mode data sets. In Figure 2, the general trend seen is an increase in classifier accuracy when only using the extracted features in the classification model. The effectiveness of the method seemed to favour certain classification problems, such as the LP4 data set, where the classification accuracy improved by as much as 42%. However, in the LP5 data set there were mixed results in the improvement of the classifier; in one run for example, the accuracy decreased by 18% when using the extracted features.

The methods described here are very sensitive to the spread of the data, and will struggle with classes that have features that are subsets of another class, see Figure 1(a). One could attempt to overcome this limitation by applying a relevant kernel function to the data [24].



(a)



(b)

Fig. 3: The confusion matrices of the predicted and actual classes with the test data set for one of the runs with the LP1 data set. (a) Using all 90 features in the classification model to achieve an accuracy of 69%. (b) Using the three extracted features in the classification model to achieve an accuracy of 88%.

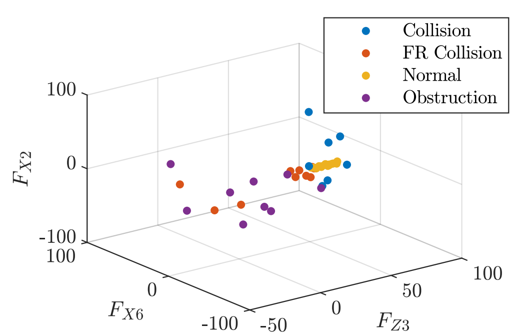
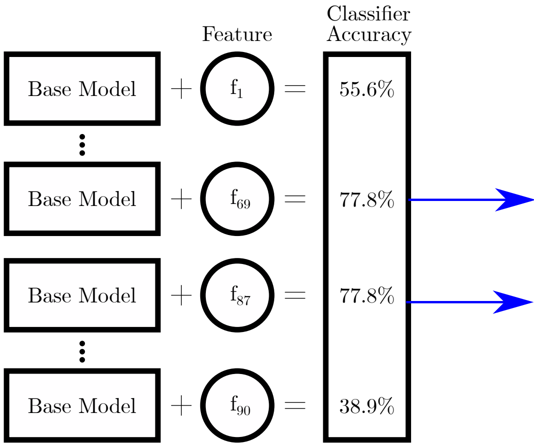


Fig. 4: Shown in the scatter plot are the features F_{X2}, F_{X6}, F_{Z3} from the training partition of the LP1 data set. These features were extracted using the method and were able to correctly classify the validation partition with 100% accuracy. When using these features for the test data, the accuracy was found to be 80% compared to the 65% when using all features.



Features that Best Improved Model
Determine What Feature to Add

Class 1	Class 2	Incorrect	Overlap	CV 1	CV 2	Score
Collision	Normal	1	N/A	N/A	N/A	N/A
Obstruction	Collision	2	0.0111	1.0803	0.0562	0.0007
Obstruction	FR Collision	1	N/A	N/A	N/A	N/A
Collision	Normal	1	N/A	N/A	N/A	N/A
Obstruction	FR Collision	1	N/A	N/A	N/A	N/A
Obstruction	Normal	2	0.0144	1.1521	0.0700	0.0012

Fig. 5: The proposed method visualised using the LP1 data set. In the first step of the process each feature is independently added to the base/null model, and the accuracy tested with the validation partition. As seen in this example, two features had the best and equal increase in improvement to the model. To determine which of the two features should be used a closer look at the classification results is needed. For each feature the mislabelled pairs are identified. The pairing with the largest number of incorrect classifications is then further scrutinised by calculating the score from the overlapping index for that pair, and the coefficient of variation for the respective classes. For the candidate features the scores are then compared, and the feature with the lower score is selected to be used in the next iteration of building the model.

In a few instances, the results in Figure 2 depict scenarios where there was little to no change in the classifier accuracy when using the extracted features. While this may seem like a failure of using the method, it is important to remember that the feature space has still been reduced, as the maximum number of features in the model construction was capped. Therefore, one can use fewer features to achieve the same classification accuracy. In applications where there may be a cumulative delay in robot processing due to sensor reading frequencies, some unnecessary sensor readings could be eliminated by using the methods proposed here. Alternatively, if one were to consider using the proposed algorithm with a classifier other than discriminant analysis, the so-called curse of dimensionality can become a significant consideration, where increasing the number of features in the classifier model can harshly impact the computational efficiency.

An inherent limitation of the forward stepwise analysis based methods are that they may not find the best possible model. The method only considers the next available feature that will improve its accuracy, which can lead to approaching only a local best solution. There may exist a better global solution that the method had not considered due to a combination of features that could yield better model accuracy. This is an advantage of methods like Random Forest [25], [26], which can explore more combinations of features to build the a better performing model.

Lastly, the run-to-run discrepancies in the model improvement results indicates a dependence on the variation of values of the training data. In data sets with a smaller number of observations, an outlier sample will have a larger impact on the data metrics, which are crucial to both the discriminant analysis classifier and the feature scoring criteria. It is thus recommended to carefully remove significant outliers in the data sets prior to applying the method and to collect a sufficient number of samples to effectively train the model.

V. Conclusions

This paper applied a modified forward stepwise model building method to a robot failure mode classification problem. The proposed modification shown in this work is to use scoring criteria when faced with multiple features that could improve the model during an iteration. The supervised model building method was used with three data sets that contained force and torque measurements of a robot during normal operation and in a failure mode. The outcome of the experiment was positive, where most models developed were able to both use a small number of required features and improve the model accuracy, when compared with using all available measurements. However, some of the results did show worse performance when using the model built from the method. In the future development of the proposed model builder it is needed to investigate these failures and how the method can be improved. Only QDA was used in this paper, and the data sets may be more receptive to other classifiers, such as k-nearest neighbours or support vector machines.

While using the proposed method was successful in reducing the number of features in the model while improving accuracy, the scope of this paper did not investigate other feature reduction techniques such as Boruta [27]. Comparing the efficiency and model improvement using these different feature reduction methods on the same data is recommended.

The applicability to other real-world systems in operation is to be seen. It is crucial that the data sets be adequately labelled for these methods to function. For instance, a mobile robot may be in a failure state for a variety of reasons, it is up to the supervising creator of the database to determine an appropriate label for this failure state. Oversimplifying the labels into broad groupings may result in poor clarity of robot status, or difficulty in finding relevant features that are applicable to all states lumped in the group. On the other hand, being overly specific and increasing the number of labelled failure modes may generate inaccurate classifications due to a large amount of overlap in shared values. Furthermore, an increase in the number of classes should have a corresponding change to the number of samples in the database. For some robotic systems it may be difficult to accumulate repeatable measurements for the desired failure modes. It may be needed to then generate samples from simulation of the robot, provided that an accurate simulation model of the system exists [28].

The results of this paper appear promising as a simple way to improve the ability of a robot in determining if it has encountered an execution failure. With further refinement, robot autonomy can be robustly improved by identifying faults. These improvements to robot intelligence and make applications from mobile humanoid robots to surgery assisting robot manipulators one step closer to being realised.

References

[1] K. Bimbraw, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, vol. 1. IEEE, 2015, pp. 191–198.

[2] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The international journal of medical robotics and computer assisted surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[3] G. Steinbauer, "A survey about faults of robots used in robocup," in *Robot Soccer World Cup*. Springer, 2012, pp. 344–355.

[4] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[5] E. Khalastchi and M. Kalech, "A sensor-based approach for fault detection and diagnosis for robotic systems," *Autonomous Robots*, vol. 42, no. 6, pp. 1231–1248, 2018.

[6] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[7] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Expert system framework for fault detection and fault tolerance in robotics," *Computers & electrical engineering*, vol. 20, no. 5, pp. 421–435, 1994.

[8] B. Yan, T. Zhang, and C. Xie, "Fuzzy expert system for fault diagnosis of robotic assembly," in *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527)*, vol. 1. IEEE, 2002, pp. 445–449.

[9] R. Davis and W. Hamscher, "Model-based reasoning: Troubleshooting," in *Exploring artificial intelligence*. Elsevier, 1988, pp. 297–346.

[10] R. Isermann, "Estimation of physical parameters for dynamic processes with application to fault detection of an industrial robot," in *Safety Evaluation Based on Identification Approaches Related to Time-Variant and Nonlinear Structures*. Springer, 1993, pp. 166–187.

[11] F. Caccavale and I. D. Walker, "Observer-based fault detection for robot manipulators," in *Proceedings of International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 2881–2887.

[12] F. Caccavale, "Experiments of observer-based fault detection for an industrial robot," in *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, vol. 1. IEEE, 1998, pp. 480–484.

[13] A. T. Vemuri and M. M. Polycarpou, "Neural-network-based robust fault diagnosis in robotic systems," *IEEE Transactions on neural networks*, vol. 8, no. 6, pp. 1410–1420, 1997.

[14] A. T. Vemuri, M. M. Polycarpou, and S. A. Diakourtis, "Neural network based fault detection in robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 342–348, 1998.

[15] I. Eski, S. Erkaya, S. Savas, and S. Yildirim, "Fault detection on robot manipulators using artificial neural networks," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 1, pp. 115–123, 2011.

[16] L. Jin, S. Li, J. Yu, and J. He, "Robot manipulator control using neural networks: A survey," *Neurocomputing*, vol. 285, pp. 23–34, 2018.

[17] S. Verron, T. Tiplica, and A. Kobi, "Fault detection and identification with a new feature selection based on mutual information," *Journal of Process Control*, vol. 18, no. 5, pp. 479–490, 2008.

[18] A. Alvanpour, S. K. Das, C. K. Robinson, O. Nasraoui, and D. Popa, "Robot failure mode prediction with explainable machine learning," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 61–66.

[19] T. Guo, D. Zhou, J. Zhang, M. Chen, and X. Tai, "Fault detection based on robust characteristic dimensionality reduction," *Control Engineering Practice*, vol. 84, pp. 125–138, 2019.

[20] M. A. Efroymson, "Multiple regression analysis," *A. Ralston and H. S. Wilf, Eds., Mathematical Methods for Digital Computers*, 1960.

[21] B. Kent and C. Rossa, "Electric impedance spectroscopy feature extraction for tissue classification with electrode embedded surgical needles through a modified forward stepwise method," *Computers in Biology and Medicine*, 2021. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2021.104522

[22] M. Pastore and A. Calcagnì, "Measuring distribution similarities between samples: A distribution-free overlapping index," *Frontiers in psychology*, vol. 10, p. 1089, 2019.

[23] L. S. Lopes and L. M. Camarinha-Matos, "UCI machine learning repository: Robot execution failures data set," 1999. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Robot%20Execution%20Failures

[24] J. Arenas-Garcia, K. B. Petersen, G. Camps-Valls, and L. K. Hansen, "Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 16–29, 2013.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[26] N. L. Afanador, A. Smolinska, T. N. Tran, and L. Blanchet, "Unsupervised random forest: a tutorial with case studies," *journal of Chemometrics*, vol. 30, no. 5, pp. 232–241, 2016.

[27] K. M and R. W, "Feature selection with the boruta package," *Journal of Statistical Software*, vol. 36, no. 11, 2010.

[28] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.