Robot Failure Mode Prediction with Explainable Machine Learning

Aneseh Alvanpour[1], Sumit Kumar Das[2], Christopher Kevin Robinson[2], Olfa Nasraoui[1], Dan Popa[2]

***Abstract*— The ability to determine whether a robot's grasp has a high chance of failing, before it actually does, can save significant time and avoid failures by planning for re-grasping or changing the strategy for that special case. Machine Learning (ML) offers one way to learn to predict grasp failure from historic data consisting of a robot's attempted grasps alongside labels of the success or failure. Unfortunately, most powerful ML models are black-box models that do not explain the reasons behind their predictions. In this paper, we investigate how ML can be used to predict robot grasp failure and study the tradeoff between accuracy and interpretability by comparing interpretable (white box) ML models that are inherently explainable with more accurate black box ML models that are inherently opaque. Our results show that one does not necessarily have to compromise accuracy for interpretability if we use an explanation generation method, such as Shapley Additive explanations (SHAP), to add explainability to the accurate predictions made by black box models. An explanation of a predicted fault can lead to an efficient choice of corrective action in the robot's design that can be taken to avoid future failures.**

I. INTRODUCTION

With increased automation and use of robotics in warehouses and manufacturing environments, the need to efficiently and accurately complete tasks are of utmost importance. During grasp planning and executing trajectories by a robotic manipulator, certain situations are encountered where the robot fails to accurately complete the task at hand. Examples include unreachable poses and grasp failure. In order to increase the accuracy of the robotic manipulator operation, such conditions should be detected and intervention activities can be performed for course correction of the operation.

Studies in the past have used various model-based or neural network based algorithms to detect or predict failures and perform corrective actions to mitigate them [1], [2]. A research study by Riccardo et. al. used a partial least square (PLS) based approach to monitor any faults that may occur during robot operation [3]. The PLS-based algorithm provided a statistical approach to monitor the robotic manipulator online but offline identification of faults.

The ability to determine whether a robot's grasp has a high chance of failing, before it actually does, can save significant time and avoid failures by planning for re-grasping or changing the strategy for that special case. Machine Learning (ML) offers one way to learn to predict grasp failure from historic data consisting of a robot's attempted grasps alongside labels of the success or failure.

Unfortunately, most of the work in evaluating the performance of ML predictive models has focused on improving the accuracy of the model rather than its interpretability. This led to building more powerful and complex classifiers, known as black-box models, which make understanding of the reasons behind predictions challenging.

On its own, the detection or prediction of a failure is not enough to perform corrective actions to avoid the fault. In addition, there needs to be an explanation of the fault, so that an efficient choice of corrective action may be taken. Yuming et. al. have used self-organized critical theory to explain the internal mechanism of fault occurrence [4]. The explanation of the fault allows for a system to decide which mitigation action would be effective for a given type of fault that will be encountered. For example, for certain types of faults, automated mitigation procedures may take longer than a human-operator based intervention. This allows for a speedy recovery of the system during a given task. This process effectively uses a variable autonomy based framework such as described in the studies conducted by Manolis et. al.[5], [6].

A. Contributions

In this paper, we investigate how ML can be used to predict robot grasp failure and study the tradeoff between accuracy and interpretability by comparing interpretable ML models that are inherently explainable with black box ML models that tend to be more accurate but do not come with ready explanations. Our data-driven approach uses supervised machine learning techniques to produce a fault detection or diagnosis model for modeling a nonlinear robotic manipulator (arm with several degrees of freedom). The model learning occurs offline on past observations that were recorded from previous operations or by a simulation, and the models are able to map informative features from a dataset of previous behaviors to the likelihood of failure. Once learned, the supervised model allows classifying new unseen data, in particular, data that would be produced in real time by the robot, and thus determine the likelihood of a fault. Our experiments show that one does not necessarily have to com-

*This work was supported by NSF-EPSCoR–RII Track-1: Kentucky Advanced Manufacturing Partnership for Enhanced Robotics and Structures (Award #1849213)

[1]Aneseh Alvanpour and Olfa Nasraoui are with Knowledge Discovery and Web Mining Lab, Department of Computer Science and Engineering, Speed School of Engineering, University of Louisville, Louisville, KY 40292, USA aneseh.alvanpour@louisville.edu; olfa.nasraoui@louisville.edu

[2]Christopher Kevin Robinson, Sumit Kumar Das, and Dan Popa are with Next Generation Systems Group, Department of Electrical and Computer Engineering, Speed School of Engineering, University of Louisville, Louisville, KY 40292, USA sumitkumar.das@louisville.edu; christopher.robinson.2@louisville.edu; dopopa01@louisville.edu

promise accuracy for interpretability. Specifically, we achieve this goal by augmenting highly accurate black box model predictions with an explanation generation method, known as Shapley Additive explanations (SHAP) [7], [8]. SHAP assigns to each feature in a predictive model, an importance value for a particular prediction (local explanation) or for the overall prediction (global explanation).

Understanding each feature's contribution and their effects on robot's grasp failure would help us to understand the different types of mechanisms that lead to failure. An explanation of the fault can therefore lead to an efficient choice of corrective action in the robot's design that can be taken to avoid the failure.

II. Background

The increasing need for reliability and safety in many fields such as medicine, aerospace, robotics, self-driving vehicles and other safety-critical industries, has motivated work on detecting and identifying potential faults in a system as early as possible and planning to avoid them [9].

Optimal performance and safety can be supported by the ability to check if there is a fault in the system and determining its time (fault detection), finding which part of the system could lead to a failure (fault isolation), and identifying details about the fault such as type, shape and size (fault identification) [9].

According to [10], fault diagnosis approaches can be categorized into two techniques: hardware redundancy-based and analytical redundancy-based. Analytical redundancy-based methods include model-based, signal-based, knowledge-based, data-driven, hybrid fault diagnosis, and active fault diagnosis methods. Data-driven approaches, which our paper adopts, rely on algorithms and artificial intelligence techniques to extract information from historic data of machine performance. Their advantage stems from their versatile ability to adapt to different systems and failures without relying on explicit, possibly complex system models or knowledge. Machine learning algorithms have been applied to help robots learn how to work and make decisions based on information received from sensors. Information could be in the form of image features from cameras or positions or velocity of gripper joints [11]. Despite advances in robot control, imprecision in sensing and actuation still challenge the stability of a robot's grasp [12].

Despite the versatility of ML algorithms, the need to use best performing (most accurate) black box ML models remains challenged by their inability to explain their predictions limits their potential to guide the design of more robust systems.

One way to categorize interpretable methods in ML is to determine if the interpretability is achieved by limiting the complexity of the ML model (intrinsic interpretable methods) or by applying methods that analyze the model after training (post hoc interpretable methods) [13]. Algorithms with simple structure such as decision trees [14], and linear models like logistic regression [15] are intrinsically interpretable. Providing global explanation based on coefficient estimates by logistic regression and visualizing decision paths or extracting if-then rules from decision tree algorithms make them naturally interpretable for the users and help to understand the relationship between predictors (features) and the model's prediction. Post-hoc explanation methods extract information from learned black-box models such as ensemble methods [16] or neural networks and help us to figure out "what else the model can tell us"[17]. These explanations are achieved by learning an interpretable model on the predictions of the black box model [18], [19], perturbing inputs and analyzing the black box model reactions [20], or applying both methods [21]. Since post-hoc methods are model-agnostic and explanations are independent of ML models, they can interpret any black box model without sacrificing their accuracy power [13], [17]. Also, they can be applied to intrinsically interpretable models. Among recent post hoc approaches such as Permutation feature importance [22] and Local interpretable model-agnostic explanations (LIME) [21], SHapley Additive exPlanations (SHAP) [7], [8] are now considered leading approach in explaining black box models.

SHAP provides explanations for individual predictions by calculating the contribution of each feature to the prediction based on Shapley values from game theory. Shapley values tell us how fairly the prediction (model output) is distributed among predictors (features of the historic data) [13], [23]. Moreover, the Shapley value explanations by SHAP is an additive feature attribution method which means the explanation model is linear and easy to understand.

In this paper, we use Tree SHAP which is a variant of SHAP for tree based machine learning models such as the LightGBM Classifier [24]. Tree SHAP is fast, calculates exact Shapley values, and correctly computes the Shapley values when features are dependent [13].

A. SHAP

Since SHAP is an additive feature attribution method, the explanation is represented as a linear function which makes it more understandable for the users[13]. Shapley values explain the model's output of a function f as a sum of the effect ϕ_i that each feature has contributed to the output. Based on the additive feature attribution, the explanation model of g is defined as:

$$g(z\,') = \phi_0 + \sum_{i=1}^{M} \phi_i z\,'_i \tag{1}$$

Where M is the number of features, $z\,' \in \{0,1\}^M$, and $\phi_i \in \mathbf{R}$.

The $z\,'_i$ is equal to one if a feature being observed and equal to zero for unknown ones and the ϕ_i's are the attribution of features. $f(h_x(z\,'))$ is the mapping function that evaluate the effect of feature observation. Variable S is the set of non-zero indexes in z, $f_x(S) = f(h_x(z\,')) = E[f(x)|x_s]$. $E[f(x)|x_s]$is the expected value of the function conditioned on a subset S of the input features. SHAP values combine

these conditional expectations with the classic Shapley values from game theory to attribute ϕ_i values to each feature:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)] \quad (2)$$

where N is the set of all input features [8].

III. Methodology

To conduct our research, we train different models on grasp failure simulation data, then compare the white-box models which are inherently interpretable to black-box models which come with no explanations but have higher accuracy. Then we bridge the black box model's accuracy and the white box model's explainability by extracting local and global explanations for the black box model using the SHAP method (see Fig. 1).

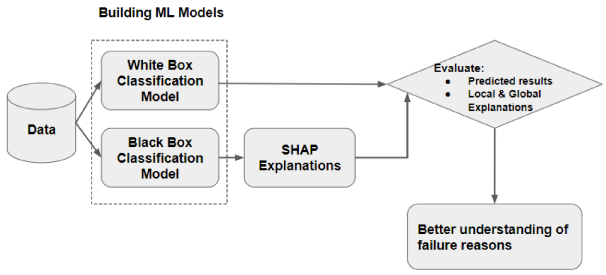

Fig. 1. Methodology flow diagram

IV. Experiments

We used a simulated robot grasp data [25] which recorded performance of a robot's arm with three fingers, including information about joints' position, velocity, effort (torque) of each finger and stability of the grasp for an object. The dataset has 992,641 records and 28 features with 448,046 records (45% of data) labeled as Stable and 544,595 records (55% of data) labeled as Unstable grasp.

As recommended in [25], we excluded the joint position from the features since the hands' shape is object-specific and we wanted to find the quality of the grasp while being object agnostic. Thus, we built our model by considering only velocity (9 features including H1-F1J1-vel, H1-F1J2-vel, H1-F1J3-vel, H1-F2J1-vel, H1-F2J2-vel, H1-F2J3-vel, H1-F3J1-vel, H1-F3J2-vel, and H1-F3J3-vel), effort (torque: (9 features including H1-F1J1-eff, H1-F1J2-eff, H1-F1J3-eff, H1-F2J1-eff, H1-F2J2-eff, H1-F2J3-eff, H1-F3J1-eff, H1-F3J2-eff, and H1-F3J3-effl) and grasp quality (one feature which is the output or target label, taking the value of Success or Failure). In total, this amounts to 18 input features and one output feature (to be predicted).

Then, we split the data randomly into training set to learn the model (794,112 records), validation set for model hyperparameter tuning (99,265 records), and test sets for assessing and reporting the generalization of the model (99,265 records). We chose the logistic regression and decision tree (DT) classifiers as white box models and the LightGBM (gradient boosting classifier) and the Multi-Layer Perceptron classifier as black box models to conduct our experiments. We then built the models using the training set and validated the results on the test set. Table 1 compares these four models based on the prediction metrics obtained by cross-validation. To avoid overfitting in the LightGBM and Multi-Layer Perceptron classifier, we tuned the hyperparameters of the algorithm using randomized search, while evaluating the model accuracy on the validation set. In the following, we report the 5-fold cross-validation results of our experiments in terms of the standard ML model's performance metrics of prediction accuracy (proportion of correct classifications), area under the ROC curve (AUC), precision (proportion of true positives out of the predicted positives), recall (proportion of true positives predicted relative to all the true positives), and F1 score (harmonic mean of precision and recall). All the metrics range in $[0, 1]$ with higher values indicating better performance.

TABLE I
ML model evaluation metrics

Metric	Decision Tree (DT)	Logistic Regression	Tree Ensemble (LightGBM)	Neural Net (MLP)
Accuracy	0.79	0.74	0.83	0.82
AUC	0.84	0.81	0.91	0.90
F1	0.78	0.74	0.83	0.81
Precision	0.93	0.83	0.93	0.86
Recall	0.66	0.66	0.76	0.75

1) Model Accuracy: As we can see in Table 1, LightGBM and MLP surpassed the DT and Logistic regression classifiers in all evaluation metrics.

In the following, we chose LightGBM for further prediction and explainability analysis, since it was the top performer in most metrics.

Fig. 2 shows LightGBM's prediction results on the test set. 41,674 of the records with the true success label were predicted correctly in the success class (True Negatives) and 41,028 of the true failures were predicted correctly to be a failure (True Positive).

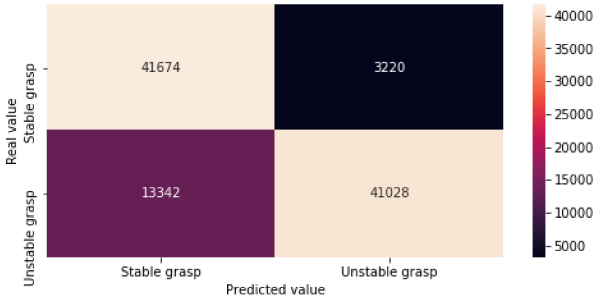

Fig. 2. LightGBM Confusion matrix

2) Model Interpretability:

a) Interpretability of the white box model, Global explanation: Important features in building a logistic regression model are achieved by calculating the coefficients of the features in the decision function. As shown in Fig. 3, joint 1's effort (torque) in finger 2 and finger 3 and its velocity in finger 3 have positive contributions in predicting the likelihood of robot's grasp failure, whereas joint 1 and joint 3 effort (torque) in finger 1 have negative coefficients contributing most negatively. This means that joint 1's effort in finger 2 and finger 3 and its velocity in finger 3 make the failure more likely and joint 1 and joint 3's efforts in finger 1 make failure less likely. Based on the estimated coefficients, the effect of the rest of features on the prediction output is very small or zero.

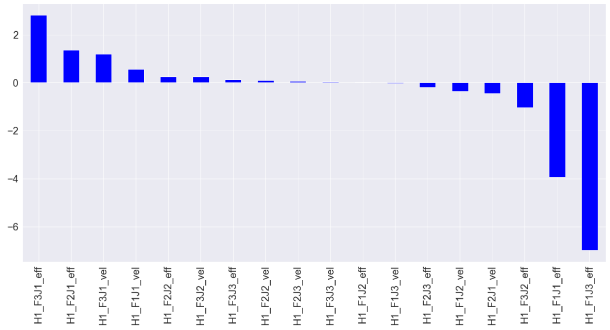

Fig. 3. Global interpretability of the entire training data by the white-box Logistic Regression

Fig. 4 also shows the features, sorted according to their importance in making decisions in the DT model. It reveals that joint 1's effort (torque) in all fingers is more important than velocity in predicting grasp stability.

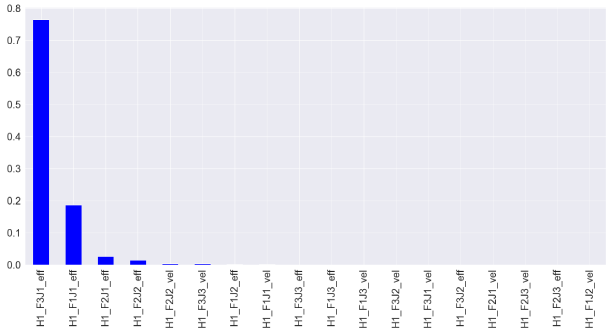

Fig. 4. Global interpretability of the entire training data by the white-box DT classifier, joint 1 effort (torque) in all fingers is more important than velocity in predicting grab stability

b) Interpretability of the white box model, Local explanation: Logistic regression does not provide local explanations for the individual data records while decision paths in the DT model show how the decisions were made. Fig. 5, displays part of these decision paths for correct classifications and incorrect classification (misclassifications). For example, following nodes 0, 32, 48, 56, 57, 59, and paying attention to the feature values that led to the final decision, we can see that all 14 records were classified correctly in the class "Stable grasp"; while the path including nodes 0, 32, 48, 56, 57, 58 resulted in 4 incorrectly classified records and 2 correctly classified records.

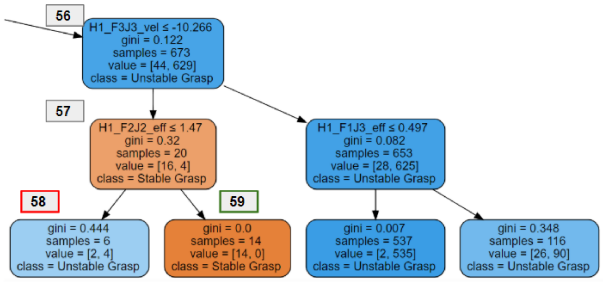

Fig. 5. The DT classifier's decision paths for correct and incorrect classifications.

c) Interpretability of the the LightGBM black box model using standard feature importance: LightGBM classifier is a tree ensemble learning method. The output of the model is considered a black box because the model consists of many individual decision trees, which are built using randomly chosen variables, thus making it difficult for users and even experts to understand the decision process.

Although global "feature importance" has been used to interpret a LightGBM model, it only gives an overview of the contribution of the features in the prediction results for the entire training data (global interpretation) and not for individual samples. Also, feature importance is not considered to be a "consistent" approach, meaning that changing the model may decrease the importance of a feature even though the feature might still have a high impact on the model's output [8].In contrast, another global interpretation method provided by SHAP has solved the consistency problem by using additive feature attribution methods and considering the attribution of the features in the output of the model. Moreover, SHAP provides global and local explanations for both training and test data sets.

According to Fig. 6, Joint 1's and 2's effort (torque) in all fingers is more important than its velocity in predicting grasp stability (failure risk in the grasp).

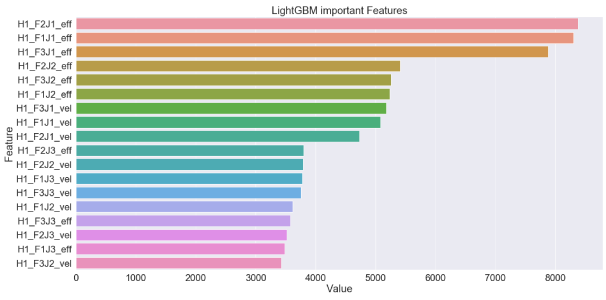

Fig. 6. Global interpretability for the entire training data by the LightGBM black-box classifier based on feature importance.

d) Interpretability of the the LightGBM black box model using SHAP values: In the following, we use the SHAP Tree explainer [8] on the prediction output of the Light-

GBM model to see what information we can extract about the robot's grasp failure cases. As we mentioned, SHAP computes two types of explanations: Local and global explanations, as we will illustrate below.

- Global Interpretability

To get a general view of which features are most important in failure prediction, we can check the global explanation plot for either training or test data set. This explanation ranks each feature based on its mean absolute Shapley value (global importance).

Fig. 7 shows that based on the Shapley values among test samples, joint 2's effort in finger 2 is the most important feature in predicting grasp failure. Joint 1's effort in the other fingers is also important as shown in Fig. 6.

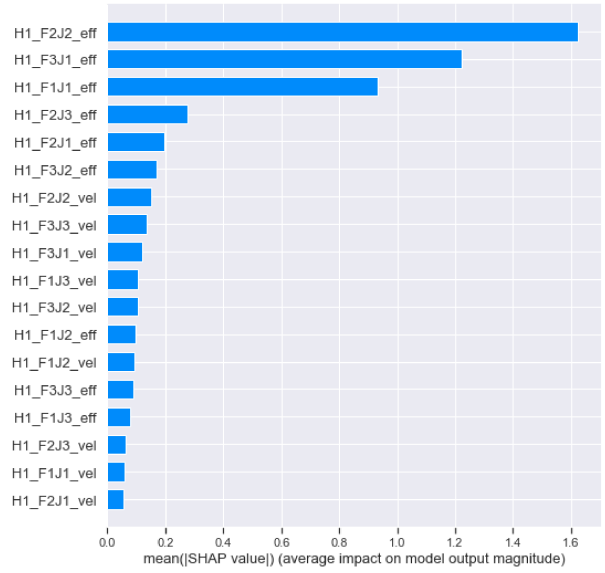

Fig. 7. Global interpretability of the entire test set for the LightGBM model based on SHAP explanations

To know how joint 2's finger 2 impacts the prediction of failure, we can examine Fig. 8 which shows the distribution and value impact of the features in detail.

To display the explanation information in Fig. 8, SHAP first sorts the features based on their global importance. Then dots, representing the SHAP values, are plotted horizontally. Each dot is colored by the value of that feature, from low (blue) to high (fuchsia).

As we can see, lower values (blue dots towards the right) of joint 2's effort in finger 2 have higher impact on the model's output, whereas higher values (fuchsia dots), have lower impact. The next important feature is joint 1's effort in finger 3 which increases the risk of failures in the robot's grasp.

- Local Interpretability

SHAP also provides explanations for any given data record (local explanation). To do that, SHAP decomposes the prediction in a graph and visualizes the feature's contribution to the prediction result. We chose one instance of true positive records from the test set and show its explanation in Fig. 9.

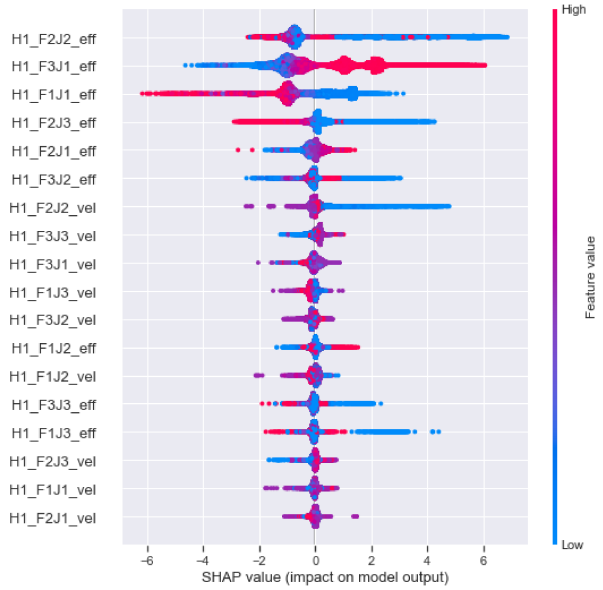

Fig. 8. Distribution and value impact of the features of the test data for the LightGBM model, explained by SHAP Global interpretability

Each feature value is a force that either increases (positive values in fuchsia) or decreases (negative values in blue) the prediction of the failure.

Our model has predicted that the robot will fail in grasping this sample of data with probability almost 1 (output value in Fig. 9). Also, the average of all predicted probabilities for the failure class in the test data (base value) is equal to 0.9038, which is the output value while ignoring all the input features.

According to Fig. 9, features in fuchsia such as joint 2's effort in finger 2 (the most important features from global explanations) and joint 1's effort in finger 1 and 3 contributed to push the model's output from the base value that ignores all features (0.9038) toward the model's actual output that takes into account the features (probability of failure for this specific record which is equal to 1).

Fig. 9. Local explanations for a true positive case predicted to be in the Failure class by the LightGBM model

Fig. 10 shows how the effort in joints' 1 finger 1 and joints' 2 finger 2 led the model to misclassify this sample of data as a success.

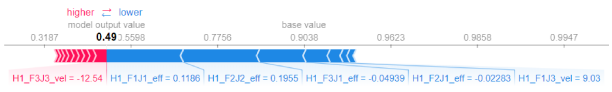

Fig. 10. Local explanations for a false negative case among the prediction results of the LightGBM model

Fig. 11 shows that while features in blue, such as joint 2's

effort in finger 2 help the prediction to be correct (decreasing the probability of failure), the effects of the fuchsia features such as joint 1's and 3's effort in finger 2 led to incorrectly classify this sample.

Fig. 11. Local explanations for a false positive case for the LightGBM model prediction

With these predictions and explanations, an alternative planning of the task can be achieved through checking the reliability of other intervention methods. For example, this can be done by utilizing a variable autonomy based framework, where the level of autonomy can be chosen for specific sub tasks such that the overall performance is optimized. In case of a prediction of a failure during a task, by observing the explanation, either the level of autonomy can be changed to manual for human intervention or an alternate path planning process can be initiated for autonomous intervention for the fault.

Our experimental results and their analysis above, illustrate the need for an accurate explanation of a predicted fault in order to be able to think about and choose an effective corrective measure.

V. CONCLUSIONS

The work presented in this paper aims at predicting and explaining the predicted fault during a robot's grasping operation. This work will be the foundation for our future work in designing a variable autonomy based architecture. The explanation of the failures, on its own, will also provide an easier avenue for the human-operator to intervene and debug the system.

Our research promises to enable a better understanding of algorithmic decision systems in robotics and to pave the way toward more trustable interactions between robots and humans.

One main limitation of our work stems from the challenges of collecting sufficient representative data, whether from real experiments or from simulations. We have used an existing benchmark simulation dataset in this paper to allow us to demonstrate, as a proof of concept, that this methodology offers a reasonable way to study "explainable" failure detection using highly accurate black box machine learning models. However future work needs to enrich the features in the data to capture more helpful contextual information about the task. Another limitation of this work is the difficulty to decipher the meaning of the SHAP visualization diagrams which require a good level of training for an expert before being able to understand and use them for decision making.

REFERENCES

[1] C. N. Cho, J. T. Hong, and H. J. Kim, "Neural network based adaptive actuator fault detection algorithm for robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 137–147, 2019.

[2] J.-H. Shin and J.-J. Lee, "Fault detection and robust fault recovery control for robot manipulators with actuator failures," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2, pp. 861–866, IEEE, 1999.

[3] R. Muradore and P. Fiorini, "A pls-based statistical approach for fault detection and isolation of robotic manipulators," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3167–3175, 2011.

[4] Y. Qi, H. Liu, B. Xie, and S. Deng, "Fault analysis of industrial robots based on self-organised critical theory," *The Journal of Engineering*, vol. 2019, no. 23, pp. 8624–8626, 2019.

[5] M. Chiou, R. Stolkin, G. Bieksaite, N. Hawes, K. L. Shapiro, and T. S. Harrison, "Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3581–3588, IEEE, 2016.

[6] M. Chiou, N. Hawes, and R. Stolkin, "Mixed-initiative variable autonomy for remotely operated mobile robots," *arXiv preprint arXiv:1911.04848*, 2019.

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in neural information processing systems*, pp. 4765–4774, 2017.

[8] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

[9] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[10] C. Cecati, "A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," 2015.

[11] S. Sapora, *Grasp Quality Deep Neural Networks for Robotic Object Grasping*. PhD thesis, Imperial College London, 2019.

[12] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[13] C. Molnar, "A guide for making black box models explainable," *URL: https://christophm. github. io/interpretable-ml-book*, 2018.

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

[15] J. S. Cramer, "The origins of logistic regression," 2002.

[16] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[17] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[18] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in neural information processing systems*, pp. 24–30, 1996.

[19] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller, "How to explain individual classification decisions," *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1803–1831, 2010.

[20] I. Kononenko *et al.*, "An efficient explanation of individual classifications using game theory," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 1–18, 2010.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[22] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, no. 177, pp. 1–81, 2019.

[23] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in neural information processing systems*, pp. 3146–3154, 2017.

[25] U. Cupcic, "Grasping dataset, shadow smart grasping system."