Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 200 (2022) 669–678

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Industry 4.0 and Smart Manufacturing

A comparative study of different algorithms using contrived failure data to detect robot anomalies

Ethan Wescoat[a], Scott Kerner[a], Laine Mears[a]

[a]*Clemson University, 4 Research Dr., Greenville, SC 29607, USA*

Abstract

Unexpected downtime remains a costly, preventable burden in manufacturing. To mitigate and eliminate unexpected downtime, manufacturers have incorporated machine learning to diagnose equipment faults and determine the equipment remaining useful life. However, such models suffer from a lack of failure data and context knowledge surrounding data gathered from production (*i.e.* rich labeled sets). The purpose of this paper is to conduct an algorithm comparison study over a previously collected contrived anomaly data set from an industrial robot. The goal of the study is to measure the effectiveness of algorithms to eliminate bias and variance from the classification results. The tested system is a 6-DOF collaborative robot from Universal Robots, on which an anomalous condition is artificially induced on a robot to simulate robot overload. The different algorithms are assessed based on their accuracy in determining the overloading case on the robot. From the analysis of data-driven machine learning and deep learning models, a deep learning regression was determined as the best model from the assessment of the data both qualitatively (low overfitting and bias) and quantitatively (98% overall accuracy). As part of the future research direction, different failure cases should be created based on wear applied to specific robot joint components and an adaptability assessment carried out for the model to other robot paths.

Keywords: Deep Learning; Condition Monitoring; Physical Twin

1. Introduction

Manufacturing robotics are proven to increase productivity by up to 36%, and to lower product output prices [1]. The productivity increase from robots has led employers to adopt their usage and take advantage of further benefits from increased safety and quality assurance in some manufacturing tasks. The increased number of robots in manufacturing represents an evolving trend towards production automation. While productivity can increase from automation, it can also introduce problems related to manufacturing adaptability, and equipment downtime [2]. Due

* Ethan Wescoat
E-mail address: ewescoa@clemson.edu

1877-0509
10.1016/j.procs.2022.01.265

to their widespread use and potential applications, robots can experience a wide range of failure types that shorten the equipment remaining useful life, leading to early failure and unexpected downtime.

Condition-based maintenance in manufacturing focuses on equipment signal assessment to determine the overall health status, transitioning previous corrective and preventive to predictive approaches. The methods vary from system to system based on the data acquisition, processing and overall analysis, as found by Jardine *et al.* [3]. However, the equipment condition determination allows maintenance staff the opportunity to correct equipment failure when it is cost-effective and before product quality degradation. Prognostic and diagnostic algorithms facilitate the implementation of predictive maintenance by providing predictions of when equipment failure could occur [4]. Maintenance staff uses the prediction to determine the following action for their equipment repair schedule.

However, production failure data for algorithm training are difficult to gather for equipment, due to the cost of failure and time until failure. Utilizing physical test beds, failure data are generated for testing new algorithms for production equipment. Qiu *et al.* utilized run-to-failure testing for developing a bearing test data set for identifying early-stage failures [5]. Lessmeier *et al.* introduced purposeful damage to bearings to generate failure data using three different damage methods to gather different damage cases [6]. Case Western Reserve University created a bearing dataset by seeding known damage on the bearing rings and rolling elements to show the incremental stages of failure [7]. Similarly to components, such as bearings, it is possible to simulate multi-component equipment failures, such as robots. Utilizing purposeful failure, Wescoat *et al.* created a robot anomalous condition to generate failure data for each robot joint data during operation [8]. By creating these failure data sets, different algorithms could be compared for determining the model prediction effectiveness in determining damage.

2. Literature Review

To ensure proper robot control, sensors are incorporated into the system structure to collect data related to equipment operations, making the data useful for condition monitoring as they provide direct correlation to robot operation. In robot condition monitoring literature, multiple different data types are collected, depending on availability, to increase the model accuracy as it relates to the robot operation and overall health condition. One method is using the kinematic and dynamic data related to robot operations. By assessing the deviation in particular movements researchers have utilized methods such as frictional wear modelling of particular robot joints [9] and power consumption modelling utilizing dynamic based data and electrical current modeling [10]. From both methods, as joint wear propagated, frictional modeling and the power consumption modeling monitor robot operation deviations from the expected values and report the robot health condition. Another data acquisition method is joint electrical current monitoring during operations to corroborate the kinematic and dynamic data collected. Yuan *et al.* conducted power efficiency estimation analysis based for a modular reconfigurable robot using kinematic, dynamic, and electrical data [11]. Yang *et al.* found the motor current data corroborates failure data trends in the kinematic and dynamic data and serves as a substitute for sensors that require placement outside the robot structure [12]. Cheng *et al.* installed additional current sensors within the robot structure to perform motor current signature analysis to determine joint wear [13]. For friction indications, Thermal data is useful as it is synonymous with wear on robot joints [14]. In addition to the sensors in the robot structure, external sensors, such as accelerometers [14, 15] and cameras [16], have been added to verify internal sensors or provide additional data streams. One example is tracking joint vibrations using accelerometers conducted by Jaber *et al.* and Eski *et al.*, who measured both vibration and noise to predict the overall vibrations on industrial robots [15, 17].

A compounding factor to algorithm development is the lack of failure data related to multi-joint operations with previous research focused on specific joint moves for the entire system [9, 12] or based on joint move simulation [18]. As mentioned previously, failure data generation has occurred with bearings to investigate degradation over the equipment lifetime using run-to failure testing [5] or purposefully induced damage [6, 7]. The bearing purposeful destruction allows for accelerated testing to gather failure data useful for predicting the remaining useful life for other equipment. As has been carried out with bearings, complex systems should be investigated in a similar manner, with the identification of a particular fault/failure and re-creation in an offline test cell. Fig. 1 shows a framework for the use of contrived data in training digital models, primarily as a means to allow for full system destruction without detrimental effects to the production line. In the comparison of these algorithms, physical testing, in the form of generated physical damage, provides a quicker path in algorithm training with representation of known damage that can be linked back to production equipment.

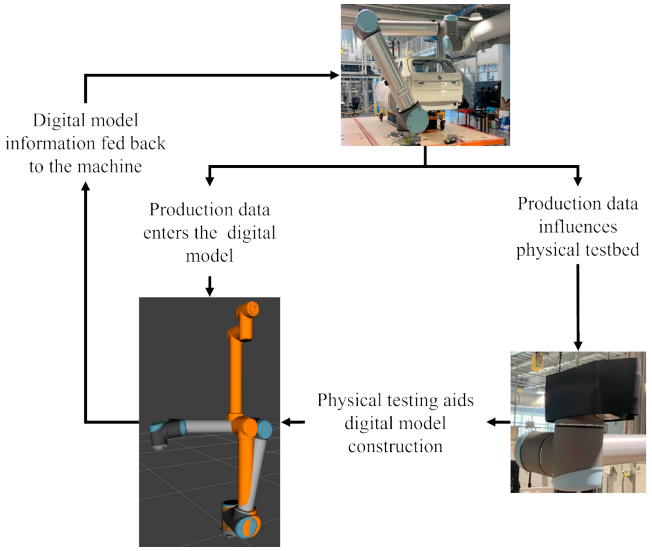

Fig. 1. Framework for integrating physical testing into digital twin data pipeline

To assess the data and test different algorithms, this paper seeks to compare the Random Forest Regression (RFR), used in the collection process, to a Support Vector Regression (SVR) and a Deep Neural Network Regression (DNNR) using contrived damage collected in a previous experiment [8]. These algorithms were chosen to investigate the data-driven methods for predicting the robot condition over physics-driven methods. RFR and SVR algorithms have been used previously with accelerometer data [19] and camera data with Izagirre *et al.* [16]. An RFR is a collection of decision trees that operate as an ensemble algorithm. Each individual tree computes a predicted value, with the final output as the average of all the decision trees, shown in Fig. 2. The random nature comes from the individual trees training on different random data samples and features. This training method helps ensure a lower correlation between the individual trees, providing a more robust model than if each tree were trained same data and features. An SVR defines the best decision boundary as a best fit hyperplane through all the data points within a certain threshold defined by the user. With the SVR, this threshold parameter requires tuning and kernel.

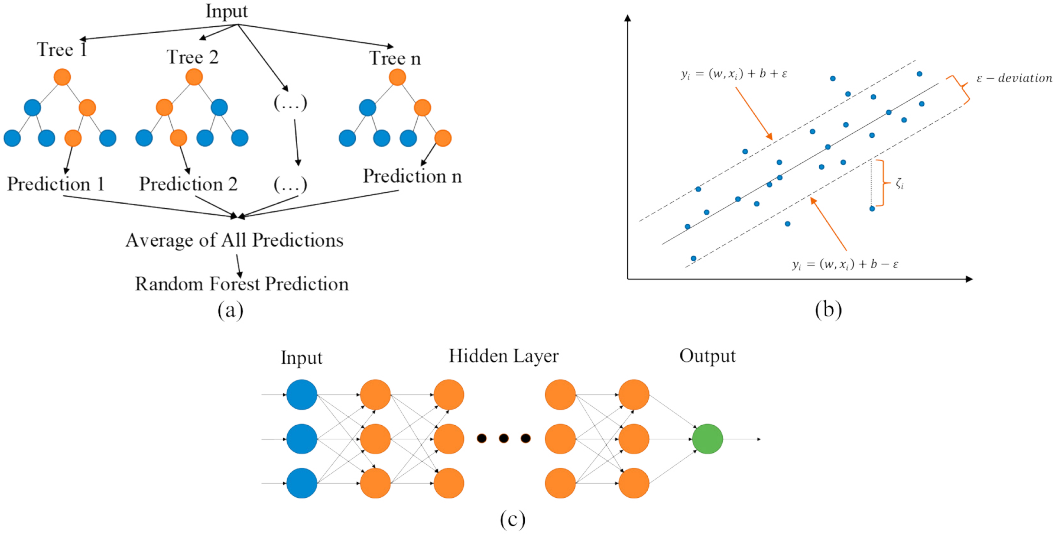

Fig. 2. Algorithm structure for the algorithms compared in this paper: (a) Random Forest Regression, (b) Support Vector Regression, and (c) Deep Neural Network Regression

Artificial Neural Networks are modeled based on biological neural networks that are found in the brain, meant to reinforce learning over time by improving results during the training period. Artificial Neural networks are comprised of three main layers: the input layer (blue), the hidden layers (orange), and output layer (green) as shown in Fig. 2. For artificial neural networks, nodes are formulated into each of these layers and are connected to each other. As training occurs, the nodes will change how the data is transformed and reach the predicted output. Vemuri *et al.* and Eski *et al.* used a neural network to determine the anomalies from the robot control state by modeling joint uncertainties [14] and measuring the vibration of robot joints for determining faults [17].

3. Materials and Methods

Section 3 is divided into two sections discussing the different algorithm structures and a brief data description. In the Algorithm Structure section, the different hyperparameters tuned for each algorithm are discussed, along with the training environment and model structure used for each algorithm. In the Data Explanation section, a brief data explanation is given with the experimental process of how the data were gathered is given.

3.1. Algorithm Structure

The three algorithms have different hyperparameters, the variables used to control the machine learning process, that are tuned during the testing phase, which are based on a random search and stopping criterion. For the RFR, the tunable parameters investigated were the number of estimators (trees), the number of features per tree, and the number of levels in each tree. For the SVR, the tunable parameters configured were the regularization, gamma, and stopping tolerance (margin). The kernel was not considered a tunable parameter, since a linear kernel was used, chosen based on prior statistical testing [8]. As such, this parameter was kept constant during testing. For the DNNR, the tuning parameters considered were the number of epochs, the number of nodes per layer, the activation function, and the number of layers. The models were created based on the software and structure provided by Scikit-learn and Tensorflow [20, 21]

Each algorithm was developed in a Python environment through Jupyter Notebook and executed on a desktop with an i7-5820K 6 core processor, 32 GB of RAM, 980 Ti GPU, and Windows 10. The RFR was designed to assign a health value associated with the entire robot, based on data from each joint in one model configuration, similar to previous work [8]. Wescoat *et al.* noted there was overfitting from training; hence, the SVR was used to determine if the problem overfitting came from the data or the algorithm. The DNNR proposed was designed to provide a better understanding of the key points during the robot operation, by using a two-tier model configuration as shown in Fig. 3. The first-tier of the model took the raw data input specific to each joint and, returning a joint health index value. The joint health index values collected determine the overall robot health. The joint health values determined where specifically the mass deviations affected the robot during operation. Another reason was to see if the change in the model configuration potentially improved results by removing the data dimensionality.

Overall robot health is defined by how accurately the model predicts the mass deviation of the robot. The mass deviation at the Tool Center Position (TCP) can cause failures over time, due to improper control of the robot operation. The improper robot operation can lead to poor product quality, in addition to early equipment failure [22]. The model accuracy was calculated based on how closely the mass predicted matched the actual mass on the end effector. The different models were assessed based on the training accuracy versus the testing accuracy to determine bias and variance for each model. The nominal error range was set to 0.1 kg for the mass limits for early model tuning. The desired error limit though is 0.01 kg since this is the robot controller accuracy range for the end effector used in these experiments. The loss was calculated during the building of the DNNR using the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). Based on the valuation of these configurations, the effective model is chosen. Section 3.2 details how the data were gathered.

3.2. Data Explanation

The data were collected from a UR10, selected based on the availability and prior knowledge in programming. The UR10 is a collaborative robot, built by Universal Robots, designed to work in close proximity of workers and handle

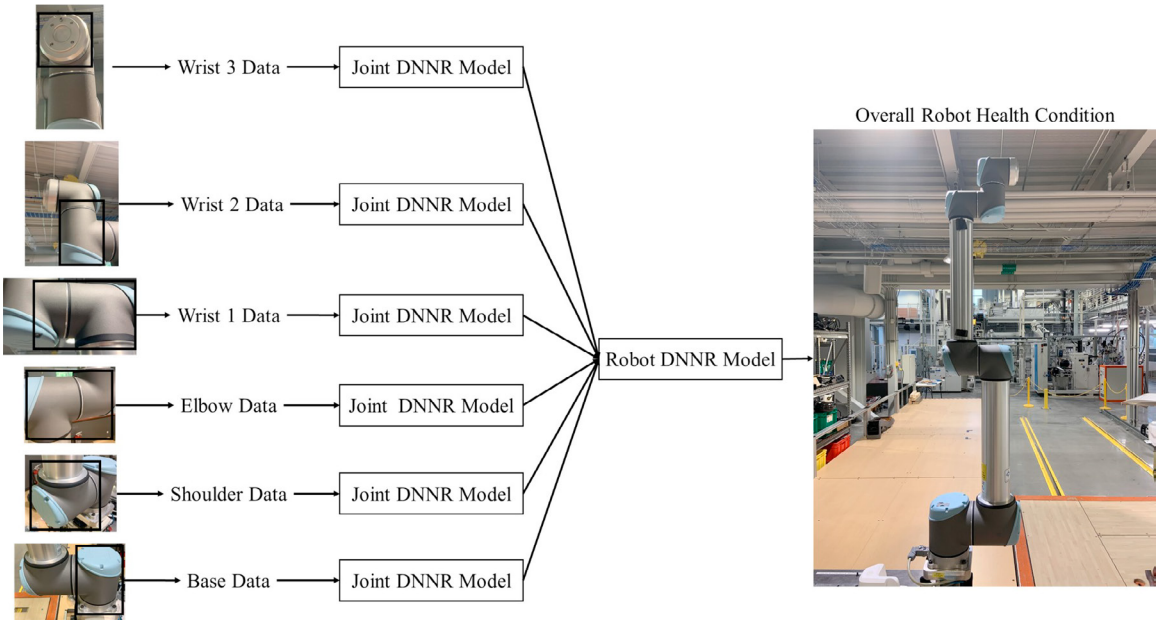

Fig. 3. Two tier model architecture for the robot health model using the joint data

payloads of around 10 kg. A couple of the prevalent use cases of the robot are in pick and place applications and quality inspections. These use cases were kept in consideration, when designing the tests for generating the data set. Fig. 4 shows the robot in the home configuration with the joint labeling. Joints 1-3 represent the robot arm (Base, Shoulder, and Elbow) portions and the large movements to reach points, where fine control becomes necessary. Joints 4-6 represent the robot wrist (Wrist 1, Wrist 2, and Wrist3) portions and the end effector fine control positioning. The interface with the robot was conducted through a teach pendant and using Robot Operating System (ROS). ROS is an open-source set of software libraries and packages that assist in robot programming. The data generated in this experiment were meant to simulate a robot probing a position close to the vehicle. The operation was run for a set number of times within each experiment. In between experiments, weights were added to the robot to simulate overloading on the end effector. The addition of weight without accounting for it in the robot control structure created a TCP misconfiguration, a potential cause of failure as cited in UR10 documentation [22].

The data were generated from six experiments, each containing one operation type that was repeated five times each with 7000 data points collected each time. The experiments consisted of a baseline test with no weight on the end effector, then adding weight in increments of 0.2 kg until 1.0 kg is reached for experiment 6. The collection occurred via sampling from the robot controller directly using ROS. During operation, velocity, joint position, current, and temperature data were collected as discussed from the prior literature in Section 1. Table 1 displays the specific data types gathered initially and their units [8]. ”Actual” data refers to data gathered which represents the true robot values collected during operation, whereas the ”Target” data represents what the controller believes the robot should be executing. With the TCP misconfiguration, these values should deviate from each other, thereby representing the overloading effects from the end effector. ”Gathered” data indicates that the data came directly from a sensor, whereas ”computed” data relates to data that is calculated from physical models with the gathered data as inputs. The joint dynamic target and actual data are computed based on the inverse kinematic equations related to linked robot joints. Similar data were collected in a production data set utilized by Vallachira *et al.* minus the electrical data [23]. While the robot motion incorporated multiple joint movements, only specific operations were considered, rather than a random operation. This method was used to determine model feasibility before incorporating multiple operations.

An initial data assessment included physical and statistical observations to determine if damage was present [8]. The physical observations investigated variations in the robot kinematics and dynamics as the anomalous condition changed, and statistical observations determined whether there were changes in the data distribution. A lack of variation in the temperature and voltage data led to those two data types removed from the overall data set. A possible reason for the lack of variation is the extent of damage of the robot and insufficient number of test runs of the data.

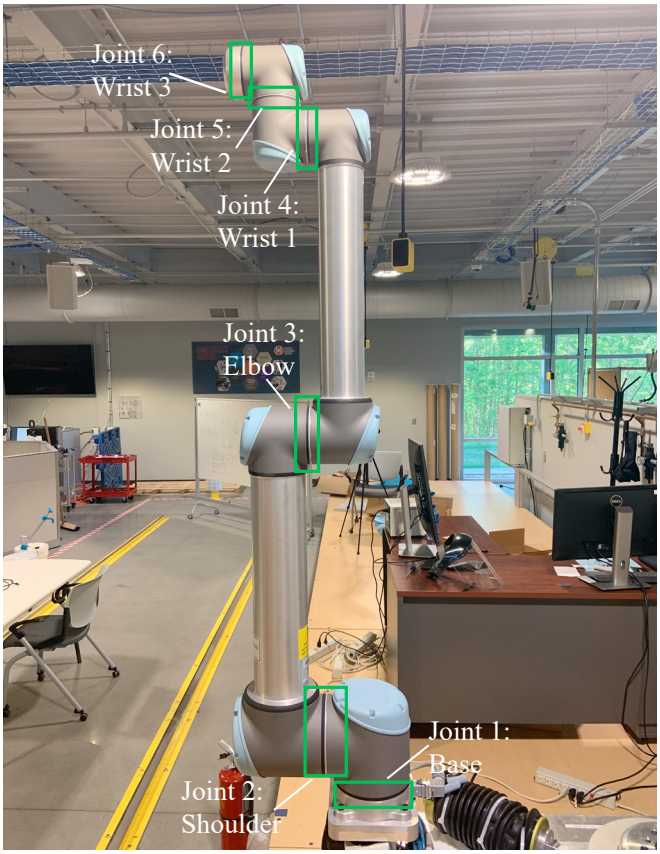

Fig. 4. UR10 showing joint names and locations

Table 1. Breakdown of collected data

Variable name	Type	Collection method	Units
Angular Position	Target/Actual	Gathered	rad
Angular Velocity	Target/Actual	Gathered	rad/s
Electrical Current	Target/Actual	Gathered	A
Torque	Target/Actual	Computed	Nm
Force	Target/Actual	Computed	N
Voltage	Actual	Gathered	V
Temperature	Actual	Gathered	C°

Subsequently, the kinematic data, the dynamic data, and the electrical current data were used to extract features for the machine learning analysis using the RFR [8]. One problem of using the RFR was the presence of over-fitting, even after algorithmic tuning and despite the data verification of the anomalous points in the data from physical and statistical observations.

4. Results and Discussion

In the RFR hyperparameter tuning, 1000 different estimators were used, a minimum number of features at 10 was configured, and the maximum depth of each tree was set as 3. For the SVR hyperparameter tuning, the regularization factor was configured as 1, the gamma was set to scale with the number of samples, and the stopping criterion was set to 0.0001. Each model was tested using a random searching initially to define the model parameters and then a grid search method was used. The stopping criterion used with the grid search was when the 5 models had a similar accuracy within 1% of each other or once 100 models had been tested with the best 5 chosen to isolate the search region. Table 2 shows the cross-validation of the final chosen models after the hyperparameter tuning, assessing the

accuracy within different ranges as defined in the table to correspond with prior results [8]. The RFR results are similar to prior algorithm expectations, whereas the SVR is an improvement in the algorithm results across each models tested in the cross-validation. However, the overall time to train and test the SVR (approximately 60 minutes), after model tuning, was unfavorable for the size of the dataset as with the variability of robot operations, retraining would need to occur during production. In addition, the model failed to meet the accuracy tolerance of 0.01 kg for the SVR. However, the SVR was unable to reach a 97% accuracy when utilizing the accuracy tolerance of 0.01 kg for the mass deviation predictions.

Table 2. Random forest and support vector regression accuracy (training / test) verified by five-fold validation

Model name	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
Random Forest Regression Accuracy (tolerance 0.1 kg)	0.98 / 0.85	0.99 / 0.85	0.99 / 0.86	0.98 0.87	0.98 / 0.85
Support Vector Regression Accuracy (tolerance 0.1 kg)	0.95 / 0.97	0.96 / 0.96	0.96 / 0.97	0.96 / 0.96	0.95 / 0.98

After training and testing the SVR and the RFR, the DNNR was trained similarly for hyperparameter tuning. Since a note of overall time was considered for the SVR, training and testing time was approximately 30 minutes for each DNNR model and 60 minutes for RFR models. The DNNR used a similar stopping criterion for model training as the RFR and SVR. The final DNNR configured model is shown in Fig. 5. There are six hidden layers, each containing 64 nodes. Originally, a lower amount of nodes were tested; however, this led to models with a higher bias and variance of accuracy in the *k*-fold cross-validation. 64 nodes was the first limit in which the five-fold cross-validation was satisfied in terms of the accuracy and the loss metrics. The activation functions are interchangeable in each layer. The Sigmoid function performed more effectively than the ReLU function in prior literature and in models with fewer layers [14, 17]; however, there are some inherent problems with the sigmoid in the DNN architecture related to the vanishing gradient principal and difficulties with convergence. Hence, different combinations of the activation functions were tuned separately, until a similar stopping criterion was met as described above. The same model was used across each joint to determine if it was possible to formulate a generic model overall.

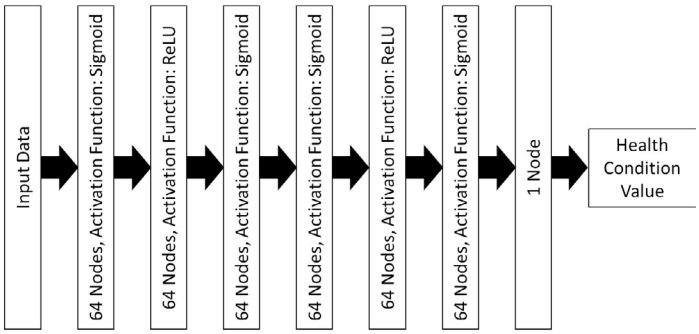

Fig. 5. Optimized DNN model

Fig. 6 shows the average prediction error for each sample in the test data set for each joint model. Each graph represents the average model results for each joint with the x-axis being the deviation between the predicted and test values. In addition to Fig. 6, Table 4 quantifies the accuracy of these models within different tolerances. The accuracy is defined as a percentage, where 100% means the model predicted the mass deviation every time. One expectation prior to modeling was to see a higher accuracy in the models related to the arm joints versus the wrist joints, since the arm joints generally experience a greater motion during operations. With the prediction error of 0.01 kg, three joints have an accuracy greater than 95%: the Shoulder, Elbow, and Wrist 1. When the prediction error is increased to 0.05 kg, there are four joint models with an accuracy greater or equal to 90%: the prior 3 joints and Wrist 2. The Base and Wrist 3 joints have the lowest accuracy, possibly that due to the fact that they experience the least movement during the actual operation. It is possible that with a higher usage during the operation, these joints may experience more of the effects from the overloading.

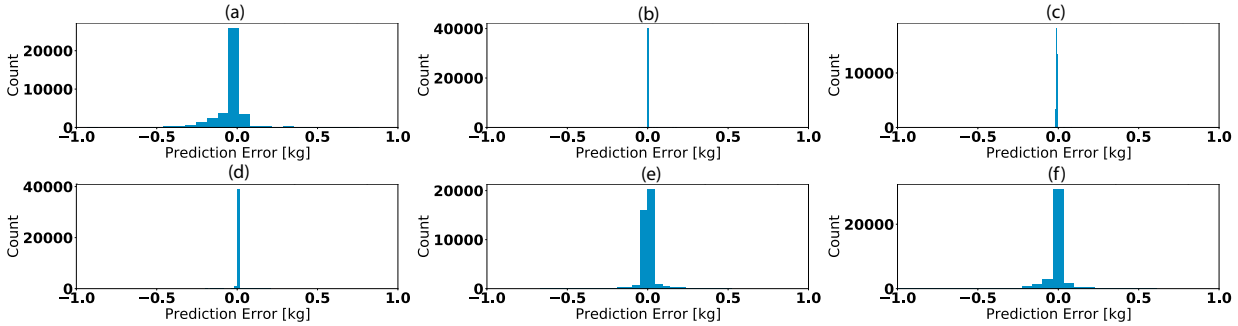

Fig. 6. Optimized joint level model accuracy demonstrating the prediction error range for each sample in each respective joint (a) base (b) shoulder (c) elbow (d) wrist 1 (e) wrist 2 (f) wrist 3. Count represents the sample count within bins of 0.025 kg.

The DNNR algorithm was used to predict the robot overall health condition utilizing the same considerations as the joint level algorithm training. The model inputs were the generated health condition values from the joint models in the prior testing. Fig. 7 shows the prediction error for the average model from each separate cross-validation conducted. The accuracy for each model was determined as: 98%, 98%, 97%, 99%, and 98%. From the cross-validation conducted, the test accuracy was assessed against the training accuracy (99%, 96%, 97%, 99%, 98%). The training accuracy error exhibited comparable testing accuracy error, signifying that overfitting potential was reduced compared to the RFR. The significance is that the current model structure is able to accurately predict the extent of the anomalous condition (the amount of mass on the end effector) within 0.01 kg of the total anomalous state 98% of the time. Another element of this framework is that the joint level model configurations helped act as ensemble learning for the overall robot health assessment, creating a higher probability of accurately predicting the anomalous state. To further prove that overfitting was eliminated, additional data should be collected of both the same operation, and different operation types to see if the same accuracy and variation occurs.

Table 3. DNN model table for each joint within a certain prediction error

Mass Error Limit	Base	Shoulder	Elbow	Wrist 1	Wrist 2	Wrist 3
0.01 kg	62%	99%	95%	97%	80%	70%
0.05 kg	75%	99%	99%	98%	90%	85%

The results of the two-tier joint model are comparable to Nentwich *et al.* and Eski *et al.* [17, 19]. Nentwich *et al.* used vibration data of two accelerometers which collected data as the robot rotated its second joint. The data-driven approach produced results which were comparable to the joint level modeling, but did not consider the robot overall health. The data taken related to only one particular joint move during operation. Eski *et al.* utilized accelerometers to compare the joint vibrations to the robot physical model. They were able to find close relation as well between the data-driven modeling with the overall physics-based changes. The two-tiered model DNNR model incorporated sensors already built into the robot structure and may need to consider additional data streams to determine similar comparisons. The DNNR model did have good accuracy overall with the data-driven approach of using a two-tier model configuration. Another similar approach that would benefit the tuned deep learning models, such as the DNNR, includes the data-driven statistical measure work to discover robot manipulator accuracy errors for predictive maintenance, as conducted by Borgi *et al.* [24]. The statistical measures can be strengthened by deep learning or become a precursor to identifying critical components in the machine with less computing power.

5. Conclusions

Different machine algorithms were compared against each other to determine their effectiveness in analyzing potential robot faults. The dataset for model training was collected utilizing a contrived failure methodology for generating failure data for production equipment. The dataset collected kinematic, dynamic, electrical current, and temperature data from six experiments in anomalous conditions. The anomalous condition was an overloaded end-effector to affect the robot dynamics and path planning. Each experiment represented a different amount of overload to determine the overload extent for an actual failure condition.

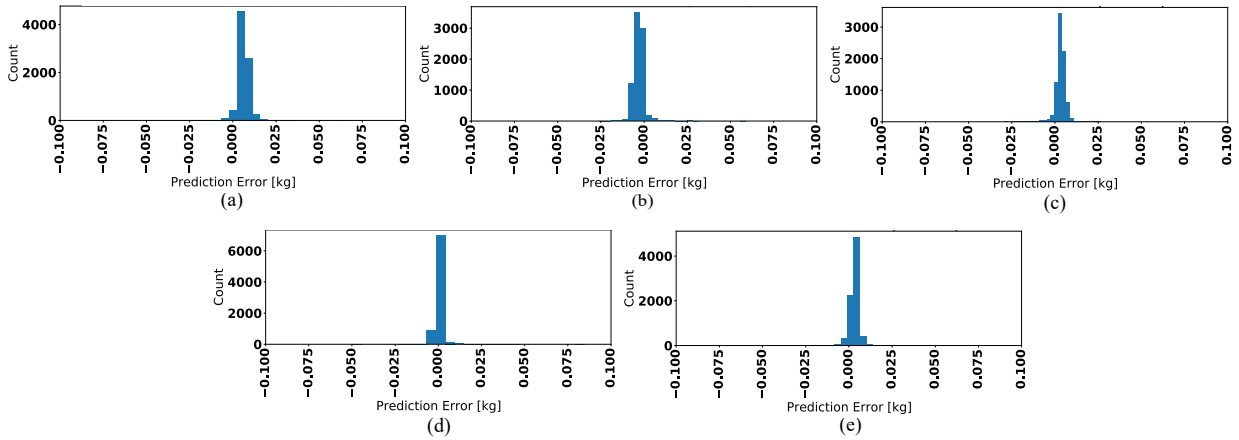

Fig. 7. Overall robot anomalous conditions for a five-fold model accuracy (a) fold one (b) fold two (c) fold three (d) fold four (e) fold five. Count represents the sample count within bins of 0.0025 kg.

The data were initially processed using MATLAB and sorted based on the experiment and respective joint. The algorithms compared were: a Random Forest Regression, a Support Vector Regression, and a Deep Neural Network Regression. The Random Forest Regression had a nominal training accuracy of 98%, but a nominal testing accuracy of 85%, falling in line with results from [8]. A Support Vector Regression was utilized in addition and had a nominal training and testing accuracy of 95% - 98%. The testing accuracy improved and removed the potential of overfitting from the RFR; however, the model accuracy was not as high as other robot papers utilizing deep learning techniques.

A Deep Neural Network Regression was used to determine if it could increase the accuracy to comparable results of other papers. A two-tier regression model assigned health variables to both the robot joint and the overall robot health. The robot joint models predicted four joints accurately the correct anomalous state 90% or greater (within 0.05 kg of the overload condition). For the two other joints, the potential model error could stem from the lack of movement from the robot operation. The robot joint health values determined the overall robot health using another DNN, with a test accuracy of 98%. The training and validation accuracy were 99% and 98%, respectively, indicating the potential for low bias and overfitting. Table 4 contains the training, validation, and test accuracies for each model used in this work.

Table 4. DNN model table for each joint within a certain prediction error

Algorithm (prediction tolerance)	Training Accuracy	Validation Accuracy	Test Accuracy
Random Forest Regression (within 0.1 kg)	0.98 - 0.99	0.86	0.85 - 0.87
Support Vector Regression (within 0.1 kg)	0.95 - 0.96	0.96	0.96 - 0.98
Deep Neural Network Regression (within 0.01 kg)	0.96 - 0.99	0.97	0.97 - 0.99

From this work, the primary contributions stem from further assessment of data generated from utilizing a mixture of traditional data-driven machine learning, and deep learning algorithms. Using the SVR model reduced model overfitting and bias, but the model accuracy was not within the prediction tolerance of 0.01 kg. A DNNR in a two-step configuration improved the model accuracy, with negligible overfitting and bias. The DNNR helped prioritize which joints were most affected by the overload condition with joint assessments before determining the overall robot health state. The approach constitutes a hybridized approach using the joint health values to determine robot state instead of the robot dynamics and kinematics.

6. Future Work

Future work will entail collecting data following a purposeful failure methodology by introducing known damage modes simulated through potentially faulty parts to test failure responses. Other robot path data is needed to determine if the DNNR two-tier configuration model is trainable for multiple robot operations. Other testable model

configurations planned are the use of Convolutional Neural Networks and Generative Adversarial Networks to determine if they improve the accuracy at the joint and overall robot level for determining the health state.

References

[1] Graetz, G., and G. Michaels. (2018) "Robots at Work." *The Review of Economics and Statistics* **C** (**5**): 755–768.

[2] Sull, D., and C. Reavis. (2019) *Tesla's Entry into the U.S. Auto Industry*.

[3] Jardine, Andrew K.S., Lin, Daming, Banjevic, Dragan. (2006) "A review on machinery diagnostics and prognostics implementing condition-based maintenance,"*Mechanical Systems and Signal Processing*, **20** (**7**):1483-–1510.

[4] Vogl, Gregory W., Weiss, Brian A., Helu, Moneer. (2019) "A review of diagnostic and prognostic capabilities and best practices for manufacturing." *J Intell Manuf* **2019** (**30**):79-95

[5] Qiu, H., Lee, J., Lin, J., and G. Yu. (2006) "Wavelet Filter-Based Weak Signature Detection Method and Its Application on Rolling Element Bearing Prognostics." *Journal of Sound and Vibration* **289** (**4–5**):1066-–1090.

[6] Lessmeier, C., Kimotho, J. K., Zimmer, D., and W. Sextro. (2016) "Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification." *Third European Conference of the Prognostics and Health Management Society 2016*, Bilbao, pp. 1–17.

[7] Smith, W. A., and R. B. Randall. (2015) "Rolling Element Bearing Diagnostics Using the Case Western Reserve University Data: A Benchmark Study," *Mechanical Systems and Signal Processing* **64–65**(**2015**):100-–131.

[8] Wescoat, E., Krugh, M., and L. Mears. (2021) "Random Forest Regression for Predicting an Anomalous Condition on a UR10 Cobot End Effector from Purposeful Failure Data." *Procedia Manufacturing* **53** (**2021**):644–655.

[9] A.C. Bittencourt. (2014) "Modeling and Diagnosis of Friction and Wear in Industrial Robots." *PhD Dissertation*. Linkoping University

[10] Sabry, A., Nordin, F. N., Sabry, A., and M. Kadir. (2020) "Fault Detection and Diagnosis of Industrial Robot Based on Power Consumption Modeling." *IEEE Transactions on Industrial Electronics* **67** (**9**):7929-–7940.

[11] Yuan, J., Liu, G., and B. Wu. (2011) "Power efficiency Estimation-Based Health Monitoring and Fault Detection of Modular and Reconfigurable Robot." *IEEE Transactions Industrial Electronics* **58** (**10**):4880–4887

[12] Yang, Q., Li, X., Wang, Y., Ainapure, A., and J. Lee. (2020) "Fault Diagnosis of Ball Screw in Industrial Robots Using Non-Stationary Motor Current Signals." *Procedia Manufacturing* **48** (**2019**):1102-–1108.

[13] Cheng, F., Raghavan, A., Jung, D., Sasaki, Y., and Y. Tajika. (2019) "High-Accuracy Unsupervised Fault Detection of Industrial Robots Using Current Signature Analysis." *2019 IEEE International Conference on Prognostics and Health Management* pp. 1–8.

[14] Vemuri, A., and M. Polyvarpou. (2004) "A Methodology for Fault Diagnosis in Robotic Systems Using Neural Networks." *Robotica* **22** (**2004**):419-–438.

[15] Jaber, A. A., and R. Bicker. (2016) "Fault Diagnosis of Industrial Robot Bearings Based on Discrete Wavelet Transform and Artificial Neural Network," *International Journal of Prognostic Health Management* **7** (**2**):1-–13.

[16] Izagirre, U., Andonegui, I., Eciolaza, L., and U. Zurutuza. (2021) "Towards Manufacturing Robotics Accuracy Degradation Assessment: A Vision-Based Data-Driven Implementation." *Robots and Computer Integrated Manufacturing*, **67** (**2021**):102029.

[17] Eski, I., Erkaya, S., Savas, S., and S. Yildirim. (2011) "Fault Detection on Robot Manipulators Using Artificial Neural Networks," *Robotics and Computer Integrated Manufacturing* **27** (**2011**):115-–123.

[18] Cho, C. N., Hong, J. T., and H. J. Kim. (2019) "Neural Network Based Adaptive Actuator Fault Detection Algorithm for Robot Manipulators." *Journal of Intelligent Robotic Systems* **95** (**2019**):137-–147.

[19] Nentwich, C., Junker, S., and G. Reinhart. (2020) "Data-Drive Models for Fault Classification and Prediction of Industrial Robots." *53rd CIRP Conference on Manufacturing Systems*, pp. 1055–1060.

[20] F. Pedregosa, G. Varoquax, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. Scikit-learn: Machine Learning in Python. (2011). *Journal of Machine Learning Research*, **12** (2011):2825–2830.

[21] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, et al.TensorFlow: Large-scale machine learning on heterogeneous systems, (2015). Software available from tensorflow.org.

[22] Universal Robot A/S "Universal Robots UR10 Service Manual" (2019)

[23] Vallachira, S., Orkisz, M., Norrlof, M., and S. Butail. (2019) "Data-Driven Gearbox Failure Detection in Industrial Robots." *IEEE Transactions on Industrial Informatics* **16** (**1**):193-–201.

[24] Borgi, T., Hidri, A., Neef, B., and M. S. Naceur. 2017, "Data Analytics for Predictive Maintenance of Industrial Robots," *Proceedings of the International Conference of Advance Systems and Electric Technologies IC-ASET 2017*, pp: 412–417.