2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Online prediction of threading task failure using Convolutional Neural Networks

Guilherme R. Moreira[1], Gustavo J. G. Lahr[1], Jose O. Savazzi[2], Thiago Boaventura[1] and Glauco A. P. Caurin[3]

Abstract— **Fasteners assembly automation in different industries require flexible systems capable of dealing with faulty situations. Fault detection and isolation (FDI) techniques are used to detect failure and deal with them, avoiding losses on parts, tools or robots. However, FDI usually deals with the faults after or at the moment they occur. Thus, we propose a method that predicts potential failures *online*, based on the forces and torques signatures captured during the task. We demonstrate the approach experimentally using an industrial robot, equipped with a force-torque sensor and a pneumatic gripper, used to align and thread nuts into bolts. All effort information is fed into a supervised machine learning algorithm, based on a Convolutional Neural Network (CNN) classifier. The network was able to predict and classify the threading task outcomes in 3 groups: mounted, not mounted or jammed. Our approach was able to reduce in 10.9% the threading task execution time when compared to a reference without FDI, but had problem to predict jammed cases. The same experiment was also performed with other two additional learning algorithms, and the results were systematically compared.**

I. INTRODUCTION

Recent developments in manufacturing have been applied to handle a diverse number of parts and subsystems. Although the end product is quite unique to the customer, the initial processes, prior to finishing and detailing, are always repetitive. A very clear case is that of fasteners: besides being used in several basic processes, are widely used in many industries [1], with market size expected to grow to more than USD 13 billion by 2025 [2].

The reliable and productive automation of fasteners insertion demands diverse integrated knowledge, such as manipulation, control, manufacturing, and machine learning [3], [4]. Although robots must be able to handle different requirements and scenarios, e.g. different geometries and dimensions, essentially all of them require a similar process: tool positioning, fastener alignment, and final insertion. During each task the robot is subject to several sources of failure. For example, it is possible that the robots fail to position the fasteners and resulting in an insertion failure or a collision of the tool with the workpiece. Alternatively, even when the parts are correctly positioned, there are situations where the alignment is not precise enough and generates a jammed situation. This may lead to a maintenance shutdown or even damage to the tool or to the robot. For tasks that require a high number of fasteners, such as the assembly of an aircraft fuselage and oil storage tanks, failures can be even more critical and costly.

In this context, failure, diagnosis and isolation (FDI) techniques may be useful for error analysis purposes. Although FDI techniques have been extensively studied, and while it is not possible here to give exhaustive reference to the available literature, we discuss some of the most related contributions to our work. In both mobile and industrial robotics, FDI was used to detect problems during operation using either data-driven techniques, such as artificial neural networks [5], [6], or model-based techniques [7], [8]. The employment of FDI in collaborative robotics is also an increasing area of interest [9], [10].

Specifically in manufacturing processes involving manipulation, several assembly applications were analyzed using FDI. Some approaches use force signature analysis [11], which is the study of the wrench profile through time, to differentiate successful cases from failures. On the model-based category, in [12] an implementation with Bayesian time-series was proposed to detect alignment problems during the whole task, and not only when the contact is achieved. The use of formal languages and automata theory to detect force transients during peg in hole tasks was investigated in [13], while [14] studied the assembly of electrical connectors using set-membership techniques applied to switched systems.

Meanwhile, data-driven based for FDI approaches have received plenty attention. It was applied, for instance, in the detection of force transients for part alignment and insertion [15]–[17], flexible parts picking [18] and in a diverse set of dynamic tasks [19]. The threading process was also a target of data-driven approaches classifying the results and study the failure and recovery. The classification during assembly using self-tapping threaded fastenings was achieved using artificial neural networks [20]. The use of robots with electrical screwdrivers was also studied, adopting support vector machine as classifier [21]. However, these cases provide only offline analyses and classification.

In this paper an *online* data-driven Convolutional Neural Network (CNN) is proposed for failure prediction of threading tasks. We show that our approach is capable of analyzing the wrench vector while the task is executed and,

[1]Guilherme R. Moreira, Gustavo J. G. Lahr and Thiago Boaventura are with the Mechanical Engineering Department at São Carlos School of Engineering, University of São Paulo, São Carlos, Brazil. guilherme.ribeiro.moreira@usp.br

[2]Jose O. Savazzi is with GPX-Embraer, Gavião Peixoto, SP, Brazil.

[3]Glauco A. P. Caurin is with the Aeronautics Engineering Department at São Carlos School of Engineering, University of São Paulo, São Carlos, Brazil. gcaurin@sc.usp.br

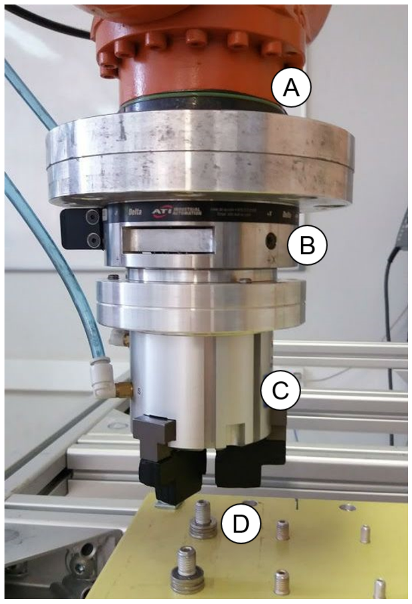
The authors would like to acknowledge CNPq and CAPES.

when a safety threshold is reached, applying a routine for recovery. For a single threading task run, it was possible to reduce the total task execution time in 10.9% with our FDI solution. In summary, the main contributions of this paper are (1) the implementation of a CNN for threading fault detection; (2) an online verification for threading tasks in robotic fastening; and (3) a quantitative and systematic comparison of our approach to other two well-known supervised machine learning algorithms, namely Multilayer Perceptron (MLP) and Support Vector Machine (SVM).

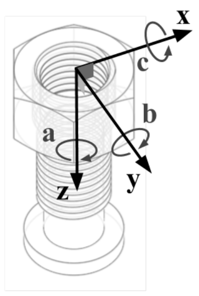
II. METHODS

A. Experimental setup

The tool is a SMC three-fingered pneumatic gripper, model MHS3-50D, with fingers 3D printed in PETG and designed to grasp all sides of a hexagonal screw nut. An industrial robot Kuka KR16 is used. The forces and torques are fed back by a 6 axis sensor from ATI Automation placed at robot's wrist. Fig. 1a displays the experimental setup, and Fig. 1b the frame used.



(a) Robot, FT sensor, gripper and bolts



(b) Cartesian axes using the nut as reference

Fig. 1: Experimental setup and axis placement: (a) detail of the experimental setup consisting of robot Kuka's end-effector (A), force-torque sensor (B), pneumatic gripper (C), above the fixed bolt (D); (b) moving frame used for the assembly task.

A Python-based code was written and used to maintain the communication between robot and computer. Positioning, impedance control and data collection are all executed by the computer via TCP/IP communication, allowed by the use of the package Robot Sensor Interface 2.3.

B. The assembly task

Our experiment consists into three steps: translational alignment, rotational fitting and fastening threading. First, the robot picks up the nut on a programmed position and takes it close to the threaded bolt. The bolt's position does not change along the experiment, but the position before the alignment is randomized via algorithm. When the nut is close to the bolt, two misalignments are inserted: a translational, through x and y axis, and a rotational, through the b and c directions. The value of the misalignment increments follows a standard normal distribution, with $\mu = 0$ and $\sigma = 0.7\ mm$ for the translational, and $\mu = 0$ and $\sigma = 0.3°$ for the rotational one. This way, we can emulate uncertainties intrinsic to an unstructured environment, closer to a real scenario. The robot then stops at a position above the bolt, with the incremental misalignment summed. Reaching this position, interaction controller is then turned on, starting the translational alignment phase. During this stage, the interaction controller changes the robot's end-effector until the desired force in z direction is reached. This phase occurs only during 20 seconds.

Fig.1b shows the axis used along the threading direction, where in z-axis a $-30\ N$ force is desired. For x and y directions we track null force. Impedance control is used as the interaction control technique, in all three translation axis of the task space. As the environment stiffness is different in z and x-y axis, the impedance control also was set different: z has parameters equals to $M = 15 \times 10^3\ kg$, $B = 200\ Ns/mm$ and $K = 5000\ N/mm$; and x-y, $M = 15 \times 10^3\ kg$, $B = 200\ Ns/mm$ and $K = 3000\ N/mm$. These parameters were chosen due to empirical experiments to reduce overshoot and oscillations. There's no corrections for the rotational fitting, due the alignment assumption, but all torque data is saved and used on the analysis.

The second step is the rotational fitting: it consists in rotating the nut in the opposite direction of threading, to achieve an accommodation of the nut. This way, the robot turns the nut for $300°$ opposite to threading direction, so the nut is capable to accommodate and is prepared to the fastening task.

Thus, the third and last step is the threading itself, rotating $600°$ to fix the nut into the screw. This setup leads to three possible situations: mounted, jammed and not mounted. The last two states are considered faulty cases and this work aims to predict them using just the data obtained during alignment. Each non-successful case can be explained by each misalignment: the jammed case (Fig. 2b) occurs when the nut is threaded at the screw with B and/or C axis misalignments causing the nut and screw threads to overlap; while not mounted case (Fig. 2c) occurs due to translational misalignment, causing the reference force at z axis to be reached without any alignment at x or y axis. Fig. 2a displays the successful assembly.

The experiment was repeated 500 times, each nut was used for 30 repetitions, and wrench data were collected for x, y and z axis. For each experiment, six force/torque data arrays of size 2560×1 were collected, corresponding to the 30.72 seconds of translational alignment and rotational fitting (the robot controller has a $0.012\ ms$ resolution). The collected data are distributed as follows: 64.4% mounted,

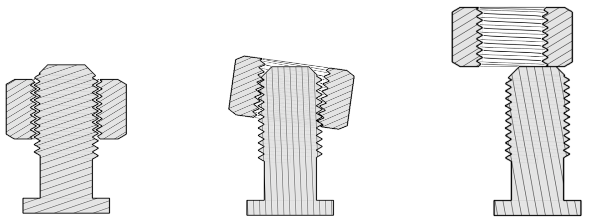


Fig. 2: Illustration of three possible assembly results. The situation where the nut is aligned but, after rotational fitting, it does not mount, was also classified as *not mounted*. In addition, the alignment phase is due to a not threaded section along the nut.

12.2% jammed and 23.4% not mounted.

The graphs in Fig. 3 shows the force and torque average at z axis of all collected data separated by assembly result, the filled area corresponds to the data standard deviation. We notice that the curve corresponding to *not mounted* data is significantly different from the two others. *Mounted* and *jammed* are very close to each other.

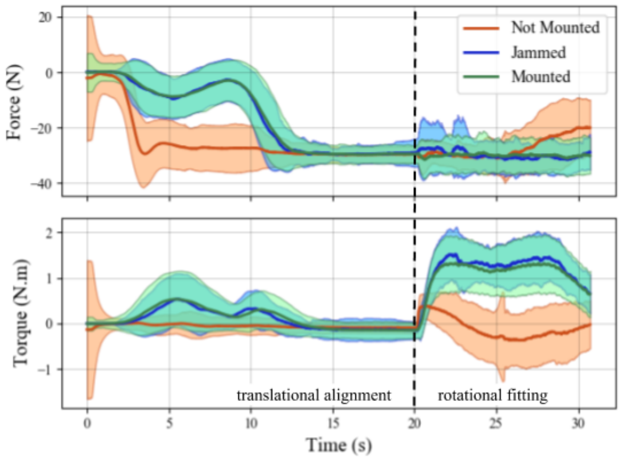


Fig. 3: Force and torque average at z axis for all collected data. Null time indicates the situation where the robot is positioned right above the bolt and is ready to align. Translational alignment is done by 20 seconds, and the dashed line indicates the starting region for the data on rotational fitting.

C. *Supervised classification and data preprocessing*

Three supervised machine learning algorithms were tested in order to classify the dynamic data: Support Vector Machine (SVM), Multilayer Perceptron (MLP) and Convolution Neural Network (CNN). SVM was configured with a Radial Basis Function (RBF) kernel and standard settings, due many FDI previous works use this method for classification tasks [15], [17], [18].

CNNs are widely used approaches to classify large amount of data, such as images or sound signals. The advantage of this method is its capability of extract features from data through convolution filters and reduce its size using pooling operations. While SVMs often needs previous features extraction, due to its high sensitivity to useless data, CNNs can extract the features by itself. In this paper, we used the architecture shown at Fig. 4 that consists in a properly scaled and standardized input layer, followed by two 1-D Convolutional with ReLU activation function plus 1-D Max-Pooling layers, a flatten operation that feeds a fully connected hidden layer with sigmoid neurons and a fully connected output layer with softmax neurons. In order to prevent overfitting we used a 0.5 dropout operation and a L2-Regularization operation ($\lambda = 0.5$) at the hidden layer. Other method used in this work is a simple MLP, that consists just by the two last layers of the CNN, that is an input layer which feeds a fully connected hidden layer followed by an output layer. The same configurations used at the CNN fully connected layers are used at MLP layers.

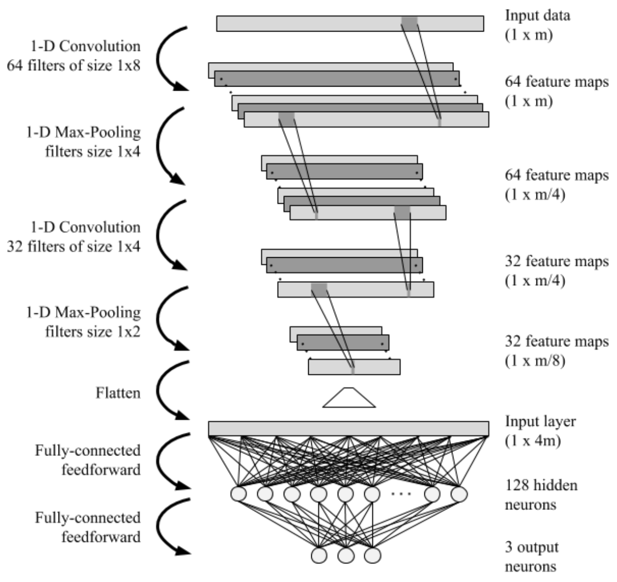


Fig. 4: Convolutional Neural Network architecture. Two stages of convolution followed by max-pooling were implemented to extract features from data. The last layers are fully connected neural networks.

To evaluate the trained models we used the *k-fold cross validation* method, with $k = 10$ $folds$. That is, for ten times each model was trained with 90% of the database and tested with the remaining 10%, generating a mean and standard deviation of the accuracies for each model. Trained models must predict the assembly result using inputs with different sizes for online prediction. To accomplish this, we will investigate the effects of two methods here called by *multi-models method* and *zero filled method*.

Using multi-models, we must train n models with input layers defined by eq. (1). During the assembly process, when the wrench vector increases its size and passes through a integer multiple $2500/n$, with $n = 10$, the network corresponding to the exact size of $2560i/n$, with $i = 1, \ldots, n$, is evaluated. If the confidence about a value is higher than a threshold of 80%, that case is taken as the predicted and the algorithm choses between stopping or threading.

$$I^{i}_{1\times(2560i/n)*6} = [F_X\ F_Y\ F_Z\ M_X\ M_Y\ M_Z] \qquad (1)$$

The *zero filled method* consists in training only one model and, while task is playing and data is increasing, the data array is interspaced with zeros in order to fit the size of the trained model input layer. In this case, the prediction input is defined by eq. (2). Both models were compared to choose the best method for online prediction.

$$I_{1\times(2560*6)} = [F_{X,1\times k}\ 0_{1\times(2560-k)}\ \ \ldots M_{Z,1\times k}\ 0_{1\times(2560-k)}] \qquad (2)$$

An important aspect of classification involving online collected data is input data standardization: all data, both used to train the network and used at the online classification, must be standardized under the same pattern. Thus, we divided all force arrays by 30 N and all torque arrays by 3 $N.m$. By doing so, we put all arrays in a similar and shorter range, increasing classifiers performance. All classification methods had the same input data.

III. RESULTS

Fig. 5 shows accuracy changing due to the amount of data used to train and verify the performance of each classifier. The classification accuracy did not present a clear inclination to increase with more data being fed, considering the analyzed interval. The most relevant perception is that standard variation, for both methods, decreased if more data was used to train the models.

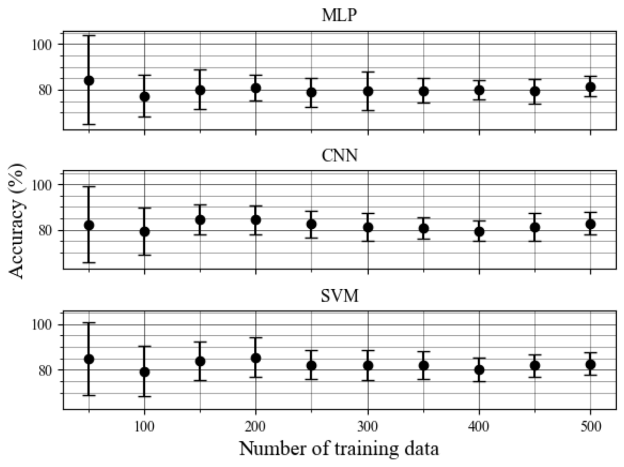


Fig. 5: Accuracy function of the amount of data used to train each classifier. Standard deviation decreases with more training data available, while the mean accuracy remains 81.6% for MLP, 82.8% for CNN, and 82.6% for SVM, with 500 training inputs, presenting a similar behavior.

Confusion matrices were created and show that none of the three classifiers were capable to predict an actual jammed situation. As presented at Tables Ia to IIIa, the classifiers had similar behavior with fake positive for mounted situation. However, the not mounted situation had different average data, and it was possible to have more accuracy on not mounted situations.

The failure of all methods to properly classify jammed cases can be explained by the great similarity in both force and torque data between correct assembly and jammed cases, as can be seen in Fig. 3. All predictions are based on data before the threading starts, using the wrench information of the translational alignment and rotational fitting. Right after these two phases, the algorithm starts the threading and, as the jamming of the nut is a specific case of the mounted situation, which the robot starts threading but the nut gets stuck. Therefore, the similarity of mounted and jammed cases during insertion is notable and hard to classify. In addition, as each nut was used 30 times in data collection, it was observed a greater tendency of jam in the last repetitions, given their wear. Moreover, even though the nuts are used only once each try, the unpredictability of a nut jam is also given by the manufacturing process of the fasteners: both nuts and bolts present manufacturing problems, resulting in a statistical relation for the case.

TABLE I: MLP metrics

(a) Confusion matrix

	$n = 500$	Actual Mounted	Actual Jammed	Actual Not mounted	
Predicted	Mounted	306	58	16	380
	Jammed	0	0	0	0
	Not mounted	16	3	101	120
		322	61	117	

(b) Outputs averages

		Actual Mounted	Actual Jammed	Actual Not mounted
Predicted	Mounted	0.80 ± 0.08	0.79 ± 0.08	0.73 ± 0.08
	Jammed	0	0	0
	Not mounted	0.71 ± 0.15	0.71 ± 0.08	0.79 ± 0.12

TABLE II: CNN metrics

(a) Confusion matrix

	$n = 500$	Actual Mounted	Actual Jammed	Actual Not mounted	
Predicted	Mounted	313	59	17	389
	Jammed	0	0	0	0
	Not mounted	9	2	100	111
		322	61	117	

(b) Outputs averages

		Actual Mounted	Actual Jammed	Actual Not mounted
Predicted	Mounted	0.81 ± 0.06	0.82 ± 0.05	0.80 ± 0.09
	Jammed	0	0	0
	Not mounted	0.80 ± 0.17	0.63 ± 0.14	0.91 ± 0.05

TABLE III: SVM metrics

(a) Confusion matrix

$n = 500$		Actual			
		Mounted	Jammed	Not mounted	
Predicted	Mounted	317	59	19	395
	Jammed	0	0	0	0
	Not mounted	5	2	98	105
		322	61	117	

(b) Outputs averages

		Actual		
		Mounted	Jammed	Not mounted
Predicted	Mounted	0.81 ± 0.06	0.79 ± 0.08	0.75 ± 0.10
	Jammed	0	0	0
	Not mounted	0.88 ± 0.11	0.63 ± 0.20	0.94 ± 0.06

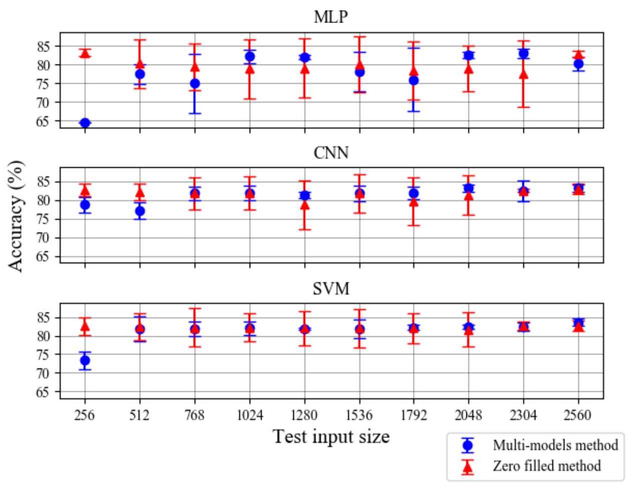


Fig. 6: Prediction performance for different input sizes of both proposed online evaluation methods: *multi-models method* (blue circles) and *zero filled method* (red triangles). Test input size corresponds to the size of the array used as input, equivalent to a time axis. Either SVM and CNN have similar performance with the mean of the predicted data, while MLP displayed significant deviation through the process.

Tables Ib to IIIb also shows the outputs averages for each classification case. This is important in order to choose correct thresholds for online classification. Considering actual mounted cases, MLP and CNN had the more useful outputs, since corrected mounted prediction had higher outputs for this case than not mounted predictions (considering only actual mounted cases). SVM give actual mounted fake positives outputs higher than real positives: 0.88 ± 0.11 for fake not mounted classification and 0.81 ± 0.06 for actual mounted classification. In this application, SVM had the worst result.

Discarding SVM as possible classifier for reasons described previously, we realize that CNN had the highest precision ($precision = true\ positives/all\ positives$) for actual mounted classifications. While CNN had a 97.2% precision for actual mounted, MLP had only 95.0% precision.

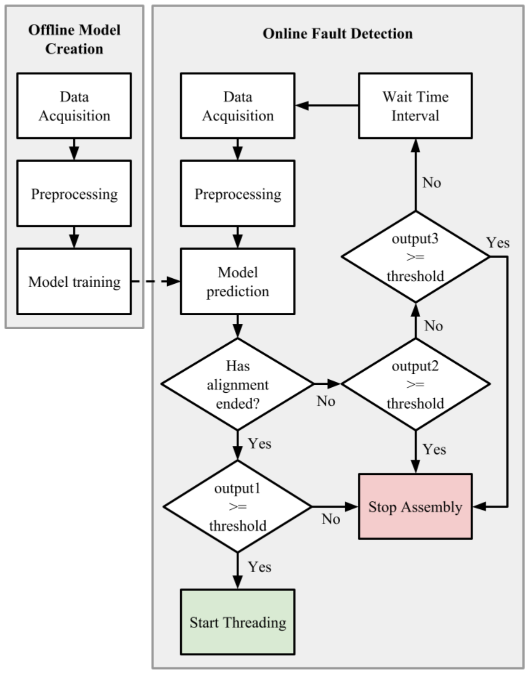


Fig. 7: Flowchart of the online fault detection proposed method. All three classifiers followed the same heuristic, with differences on the preprocessing.

Considering that mount classification is the main procedure of classification, CNN is the best classifier.

The *zero filled method* presented a higher deviation for the methods testing input, but similar performance at the final verification steps, which indicates that it could be adapted to more verification iterations, while the *multi-models method* would require more training and memory allocation.

IV. PROPOSED METHOD AND APPLICATION

The fault detection approach proposed in this work consists in two modules, shown by Fig. 7: an offline one, that contains all the procedure to train the machine learning model, and an online module, that uses the trained model to perform the fault detection while the task is occurring.

Throughout the assembly task, the same procedure is executed in equal spaced intervals with k (from eq. 2) multiple of 256. The force and torque data were collected until the last interval is preprocessed and through the trained machine learning model, resulting in three numeric outputs each one corresponding to predict values of a possible threading result: $output1$ leads to mounted, $output2$ to jammed, and $output3$ to not mounted. If threading movement is about to start, we verify if the value of the mounted state corresponding output is higher than a defined threshold. If this is the case, the task executes the threading, otherwise, the procedure ignores the threading and stops the task. In case of the alignment still happening, the system must verify if it is going to fail. To do this, the two model fault outputs are verified. If one of it is higher than a threshold, the system assumes that the task will fail and stop it.

We implemented the task with CNN classifier *zero filled method*, by repeating the technique until 30 fasteners were mounted, with the same procedure described at Sec. II-B. Considering the CNN outputs at Table IIb, we choose a threshold $t = 0.80$ for all models outputs. Each nut was used once to be as similar as possible to a real assembly situation, and to prevent fake positives at jammed situation as described previously. We measured the total time of the task to achieve 30 repetitions and the correctly mount rate of both situation. The experiment without FDI took 39 repetitions to achieve 30 mounted nuts in 77.7 minutes, while the method with FDI, 40 repetitions were necessary to achieve 30 correct in 69.2 minutes, and at 97.5% of the cases the method predicted it was going to mount and actually mounted, the only mistake was one jammed case. Although the mount rate is similar for both cases, the FDI solution decreased the execution time by 10.9%.

V. CONCLUSIONS

This work proposed a data-driven method for online prediction in threading tasks. Using an industrial robot and uncertainties in position, we simulated a real scenario condition for data acquisition and machine learning training. We've shown that with a CNN implemented and evaluation of success or failure with completion of zero values,the *zero filled method*, we could decrease 10.9% in time of a threading task. SVM had difficulty with the mounted fake positives, while MLP had less precision than the other two methods. CNN presented a good precision and less fake positives than SVM, showing as the best classifier. Analyzing other work with wrench data and CNN as the classifier [22], it is possible that with more information the CNN could present higher accuracy. Also, predicting jammed cases still a difficulty even for the CNN network, a open problem related in the literature [4]. As future research we suggest increase the database, in order to analyze impact of a larger amount of data at the used classifiers and investigate the use of Recurrent Neural Networks, which has shown good results at time series classification [23].

REFERENCES

[1] Accuray Research LLP, "Global Aerospace Fasteners Market Analysis & Trends - Industry Forecast to 2025," 2017. [Online]. Available: https://www.researchandmarkets.com/research/v5n8hr/global_aerospace

[2] Grand View Research, "Industrial Fasteners Market Size | Global Industry Research Report 2025," 2017. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/industrial-fasteners-market

[3] J. Krüger, L. Wang, A. Verl, T. Bauernhansl, E. Carpanzano, S. Makris, J. Fleischer, G. Reinhart, J. Franke, and S. Pellegrinelli, "Innovative control of assembly systems and lines," *CIRP Annals*, vol. 66, no. 2, pp. 707–730, jan 2017.

[4] Z. Jia, A. Bhatia, R. M. Aronson, D. Bourne, and M. T. Mason, "A Survey of Automated Threaded Fastening," *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2018.

[5] A. Diryag, M. Mitić, and Z. Miljković, "Neural networks for prediction of robot failures," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 8, pp. 1444–1458, jun 2014.

[6] M. Van, H. J. Kang, Y. S. Suh, and K. S. Shin, "A robust fault diagnosis and accommodation scheme for robot manipulators," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, pp. 377–388, 2013.

[7] M. McIntyre, W. Dixon, D. Dawson, and I. Walker, "Fault identification for robot manipulators," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 1028–1034, oct 2005.

[8] F. Arrichiello, A. Marino, and F. Pierri, "Observer-Based Decentralized Fault Detection and Isolation Strategy for Networked Multirobot Systems," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1465–1476, jul 2015.

[9] S. Golz, C. Osendorfer, and S. Haddadin, "Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, may 2015, pp. 3788–3794.

[10] F. Naegele, M. Naumann, and A. Verl, "A Framework for a Fault Tolerant and Learning Robotic Assembly System," *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pp. 1–6, 2012.

[11] A. Rodriguez, D. Bourne, M. Mason, G. F. Rossano, and JianJun Wang, "Failure detection in assembly: Force signature analysis," in *2010 IEEE International Conference on Automation Science and Engineering*. IEEE, aug 2010, pp. 210–215.

[12] E. Di Lello, M. Klotzbucher, T. De Laet, and H. Bruyninckx, "Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 5827–5833.

[13] Z. Jakovljevic, P. B. Petrovic, D. Milkovic, and M. Pajic, "Diagnosis of irregularities in the robotized part mating process based on contextual recognition of contact states transitions," *Assembly Automation*, vol. 35, no. 2, pp. 190–199, apr 2015.

[14] J. Huang, Y. Wang, and T. Fukuda, "Set-Membership-Based Fault Detection and Isolation for Robotic Assembly of Electrical Connectors," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 1–12, 2016.

[15] E. Krabbe, E. Kristiansen, L. Hansen, and D. Bourne, "Autonomous optimization of fine motions for robotic assembly," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, may 2014, pp. 4168–4175.

[16] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, "Detection of contact force transients in robotic assembly," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 962–968.

[17] J. Rojas, S. Luo, D. Zhu, Y. Du, H. Lin, Z. Huang, W. Kuang, and K. Harada, "Online robot introspection via wrench-based action grammars," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver: IEEE, sep 2017, pp. 5429–5436.

[18] E. Moreira, L. F. Rocha, A. M. Pinto, A. P. Moreira, and G. Veiga, "Assessment of Robotic Picking Operations Using a 6 Axis Force/Torque Sensor," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 768–775, jul 2016.

[19] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2016, pp. 407–414.

[20] K. Althoefer, B. Lara, Y. H. Zweiri, and L. D. Seneviratne, "Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 6, pp. 1081–1095, jun 2008.

[21] T. Matsuno, J. Huang, and T. Fukuda, "Fault detection algorithm for external thread fastening by robotic manipulator using linear support vector machine classifier," in *2013 IEEE International Conference on Robotics and Automation*, vol. 2, no. c. Karlsruhe: IEEE, may 2013, pp. 3443–3450.

[22] E. Roberge and V. Duchaine, "Detecting insertion tasks using convolutional neural networks during robot teaching-by-demonstration," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver: IEEE, sep 2017, pp. 3210–3216.

[23] B. H. Tan, H. Tang, R. Yan, and J. Tani, "Flexible and robust robotic arm design and skill learning by using recurrent neural networks," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, no. Iros, pp. 522–529, sep 2014.