Recurrent Neural Networks for Inferring Intentions in Shared Tasks for Industrial Collaborative Robots

Marc Maceira, Alberto Olivares-Alarcos and Guillem Alenyà

***Abstract*— Industrial robots are evolving to work closely with humans in shared spaces. Hence, robotic tasks are increasingly shared between humans and robots in collaborative settings. To enable a fluent human robot collaboration, robots need to predict and respond in real-time to worker's intentions. We present a method for early decision using force information. Forces are provided naturally by the user through the manipulation of a shared object in a collaborative task. The proposed algorithm uses a recurrent neural network to recognize operator's intentions. The algorithm is evaluated in terms of action recognition on a force dataset. It excels at detecting intentions when partial data is provided, enabling early detection and facilitating a quick robot reaction.**

I. INTRODUCTION

The continuous strive on improving the flexibility of industrial tasks is propelling the use of robots in new environments. Robots are increasingly capable of carrying more heterogeneous tasks and to perceive the environment around them [1]. Advances on artificial intelligence algorithms, together with the availability of large quantities of data, and the increased capacity of affordable processing elements, have triggered a new industrial revolution: industry 4.0 [2]. In this paradigm, robots are no longer isolated in the industry, they are in a shared workspace with humans [3]. Collaborative robots are taking the more repetitive tasks and revaluing the operator's ones. Universal Robots, the pioneer on collaborative robots, has sold more than 27000 collaborative robots around the world.

In the industry 4.0, new communication methods between humans and robots are needed to achieve a truly collaborative interaction. Robots need to react precisely and with minimal delay to the intentions of the users. In this work, we explore a collaborative scenario where a human and a robot share the execution of a task. The operator should enjoy an effortless experience, performing their actions naturally while the robot reacts to their intentions in real time.

We consider the realistic scenario proposed by Olivares-Alarcos et al. [4]. It provides an industrial collaborative task where a robot and a human share the task of cleaning and polishing an object. An example of the setup considered in this work is shown in Fig. 1. User's intentions in the dataset are captured through a force sensor which provides a natural interaction between the operator and the robot.

The baseline solution relies on machine learning approaches that require to observe a window, that is, need to obtain a significant number of samples before making a decision. The major drawback is that the classification of the intention of the user takes a significant time. Consequently, the delay of the robot's response generates discomfort to the operator and reduces the system's productivity. Thus, new early decision methods are needed to provide a quick and natural collaboration.

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain; mmaceira@iri.upc.edu, aolivares@iri.upc.edu and galenya@iri.upc.edu

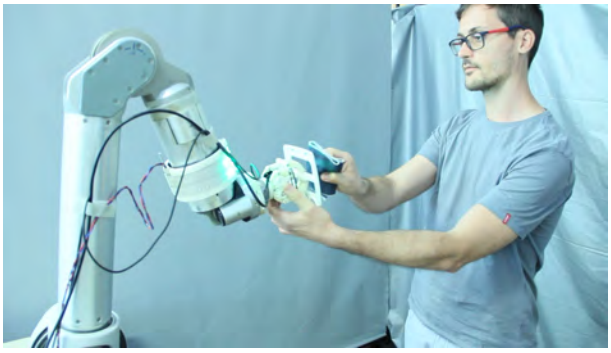

Fig. 1: Industrial scenario considered in this task. The operator can polish, grab the object or move the robot actuating over the shared object. Samples from the force sensor are used to classify user intentions, which change the robot behavior adaptively.

In this article, we use a recurrent neural network (RNN) [5] to classify user's intentions. RNNs provide two benefits compared to window-based machine learning techniques. Firstly, it reduces the decision time, as RNNs do not need a full window of measurements prior starting the classification algorithm. Secondly, it provides a closed loop system, as it provides a classification for each sensor measurement. The network is more flexible to changes, since it can react dynamically when the user changes their intention.

The main contributions of this article are as follow:

- Improved classification's accuracy and faster response time than methods in the literature in the early decision paradigm
- Continuous real time decision: Real-time intention's classification using a RNN, allowing a faster action recognition and closing the loop during task execution
- The proposed method trained with limited data still performs well, which is desirable when setting up new collaborative human-robot solutions in the industry

II. RELATED WORK

Collaborative robots are a trend in industry, but their effective use requires a thorough study of the possible interactions.

Losey et al. [6] review collaborative aspects of Physical Human–Robot Interaction. They define three key themes: intent detection, arbitration and feedback. Our work belongs to the intent detection field, where the robotic system detects what the human is trying to do from the physical human-robot interaction. Maurice et al. [7] define ergonomic indicators to estimate biomechanical demands occurring during manual activities. Collaborative robotics can improve the ergonomics of humans working with robots, in our considered use-case we consider the operator's ergonomics by allowing them to move the robot.

Communication between humans and robots range from vision, auditory, physical and other sensors or biologic signals. Among them, we are interested in physical interaction, specifically, those works where the force exchange between robots and humans is used to understand the human's actions. A good review can be found in Ajoudani et al. [8], with particular attention to our paradigm of using force/pressure sensors to determine the cooperation effort and to anticipate the objective of the operator.

Huentemann et al. [9] worked in a similar user intention recognition paradigm than ours. They used a force sensor to estimate the navigation intentions of users in a wheelchair from haptic joysticks. Gaz et al. [10] presented a robot control algorithm for an industrial task. They considered manual polishing tasks in human-robot collaboration, where the robot held an object and the human polished it with a tool. The operator communicated their intention by applying force on different parts of the robot. Thus, unlike in our work, there is no need to classify the user's intentions since each action is detected by different sensors.

Neural networks have been used also for inferring human intentions. Liu et al. [11] used haptic forces to identify desired robot's velocity based on the force applied by the human. Sharkawy et al. [12] used the velocity of the robot and the applied force to feed a multilayer neural network to modify online the virtual damping of the admittance controller. Heo et al. [13] used a Convolutional Neural Network to train a collision model for a robot using joint signals such as positions, velocities and estimated torques. In their case, they use the force sensor to label the time periods where a collision has occurred. More similar to our proposed task, Zhou et al. [14] used an RNN to predict human's actions. They worked on a surgical scenario and implemented a turn-taking prediction algorithm from multimodal data. Their objective, like ours, is to obtain an early decision method to provide faster interactions with the human.

Olivares-Alarcos et al. [4], proposed to infer the operator's intentions in order to adapt the robot's behavior while a robot and a human shared the execution of a task. Authors used a dataset containing force signals and a window-based approach to classify the operator's intention. Our work shares their objective and we will use their dataset to evaluate our method and thus be able to compare our method to theirs. Our aim is to improve the response time of the robot and the intention's inference in an early decision paradigm.

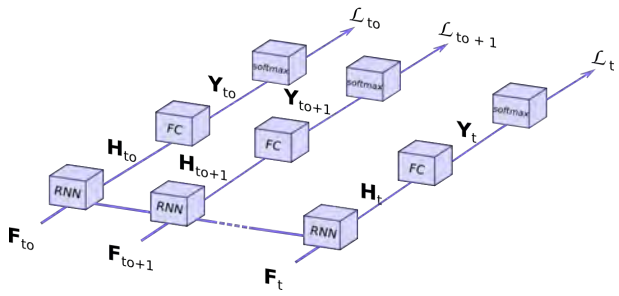

Fig. 2: Network's architecture. The network is displayed for multiple sensor measurements F_i. At each time instant the force sensor signals F_i are fed to the RNN, which updates the hidden states H_i. The fully connected layer (FC) uses H_i to obtain the user's intentions Y_i. User's intentions are converted to likelihoods $\mathcal{L}_{t_i}$ with a softmax layer.

III. RNN FOR EARLY INFERENCE OF OPERATORS' INTENTIONS IN COLLABORATIVE ROBOTIC TASKS

This section describes the algorithm proposed to infer humans' intentions in an industrial collaborative robotic task. The design of the network configuration and their characteristics is discussed in Section III-A. Once our network's architecture is defined, we explain in Section III-B the characteristics of two different decision methods that we used to infer the human's actions.

A. Selection of network's architecture

The network proposed for this task is depicted in Fig. 2. It consists of an RNN network followed with a fully connected layer. Each input sample from the sensor is passed through the network which updates the hidden states H_i. The decision of which intention is being performed by the user is done with the fully connected layer from the H_i. The fully connected layer classifies among the k possible intentions defined in the dataset, in our case, $k = 3$. Finally, the softmax layer obtains the likelihood $\mathcal{L}(t, k)$ for each of the classes. The size of the network determines the capacity to learn among the variability of the input samples. The hidden units H_i are used in the fully connected layer to detect which of the k actions is being performed at the moment. The number of hidden units delimits the complexity of the task to solve.

The use of a recurrent network architecture to infer the human's intention presents several benefits. First, it allows to naturally handle time series sequences as the recurrent architecture captures both, the instantaneous and the previous information. Hence, each sensor's measurement is feed to the network so that we get an inferred label for it. This continuous real-time decision generated by the network is able to dynamically detect a change in the intention of the user. Secondly, the neural net provides a confidence score for each measurement. This score is used latter to determine the degree of certainty of the net regarding the current action, and can be used to determines when the robot will start reacting to it. Thirdly, the RNN based approach reduces the latency

of the system. As each input measurement is processed when it is available, the latency is reduced to the inferring time.

We used two types of RNN: Long Short Term Memory (LSTM) [15] and Gated Recurrent Unit (GRU) [16]. Both allow the network to learn long term dependencies, something that regular RNN can not due to vanishing/exploding gradients.

B. Decision criteria for the human's intention inference

Since the network's architecture provides a confidence measure for each sensor measurement, we need to define a stopping criterion for the intention's classification. We propose two criteria for the decision making process to infer the human's intention k, one based on time (O_w) and the other one on confidence (O_c).

The output of the net $Y_{t,k}$ after the softmax layer provides the class' likelihood $\mathcal{L}$:

$$\mathcal{L}(t,k) = softmax\, Y(t,k) \tag{1}$$

where k indicates the class intention and t the time instant.

We define the time based criterion O_w as:

$$O_w(t_i) = \arg\max_{t_i} \mathcal{L}(t_i,k) \tag{2}$$

where t_i is the fixed decision time. The O_w criterion mimics the window-based approaches. In this case, the decision of the most probable action is done after a fixed number of sensor measurements. O_w checks the output of the net after t_i samples and takes the most probable class as the intention of the user. This decision criterion is a fair comparison with window-based methods from the literature since the same number of sensor measurements are used for each user's interaction. The first column of Fig. 3 show examples of the O_w criteria with the intention decision at a fixed t_i.

The O_w criterion does not exploit the continuous real time decision generated by the recurrent network. Therefore, we propose a confidence-based decision criterion. It consists in monitoring the output of the net for each sensor measurement and returning an inferred label when the likelihood of one actions surpasses the threshold of confidence th_c. We define the confidence-based criterion as:

$$O_c(th_c) = \arg\max_{t_i=t_{c_0}\rightarrow t} \mathcal{L}(t_i,k) \mid \mathcal{L}(t_i,k) > th_c \tag{3}$$

where t_{c_0} indicates the first time instant where the network starts monitoring the likelihood $\mathcal{L}$ to do a decision.

The O_c criterion makes the decision of which is the user's intention when the network estimates that the action likelihood is above the threshold th_c. This approach adapts the time needed for the decision depending on the complexity of the received samples. If the intention of the user is clear, just a few samples will be needed to make a decision. The setting of the threshold in the O_c criterion is a trade-off between response time and accuracy. Setting a lower threshold provides shorter latency at a cost of reducing the classification accuracy. During the evaluation we will fix different threshold values to explore the time and accuracy trade-off. The t_{c_0} parameter ensures that the network observes a minimum of sensor measurements before having the hidden states updated and start doing informed decisions. Examples of the O_c criteria are shown in the second column of Fig. 3. On the topmost example, after few samples the network is highly confident about which action is being performed by the user. Below we see an example of the need of setting the t_{c_0} parameter before the hidden states are updated. Notice that with a lower confidence threshold than the one we used (0.9), the second sequence would have been detected as class 1 instead of class 2.

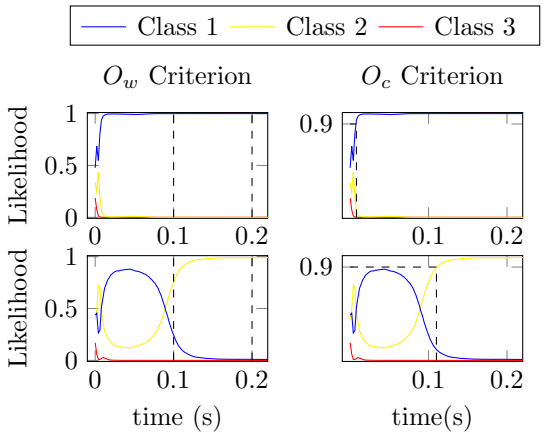

Fig. 3: Identifying user intentions. Network output: likelihood of the three possible intentions across time. In the window-based approach (O_w) the action is decided at fixed positions t_i depicted in the x-axis with discontinuous lines. In the confidence based approach (O_c), the intention is decided when the likelihood of one action surpasses a certain threshold. In this example the threshold th_c has been set to 0.9.

IV. Evaluation results: inferring the operator's intention for industrial collaborative robotic tasks

In this Section, we provide a thorough evaluation of the proposed architecture. First, we present the evaluation setup used in the experiments in Section IV-A. Second, we evaluate the network's architecture with the O_w criterion using LSTM and GRU units to analyze the performance of the net under different configurations (see Section IV-B). Third, from the configurations evaluated before, those with the best performance are compared with previous methods in the literature in Section IV-C. Fourth, the use of the confidence-based O_c decision method is discussed in Section IV-D. Finally, we analyze the degradation of the method when fewer data is used to train it in Section IV-E.

A. Evaluation Setup

The dataset used in this work [4] consists of the force/torque signals recorded from the physical human-robot interaction during the execution of a collaborative task,

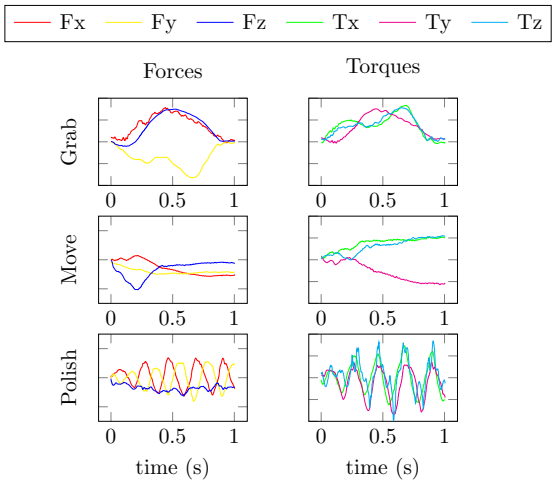

Fig. 4: Force/torque sensor data. The used dataset provides 6 signals: 3 forces and 3 torques axes. Each sequence is annotated with one of the three possible intentions of the user: grab, move or polish the object.

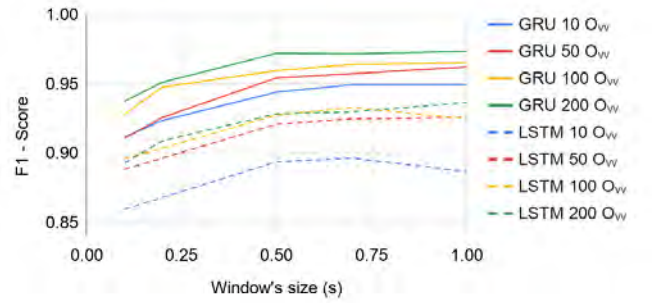

Fig. 5: RNN results: network's architecture evaluation. LSTM and GRU configurations are tested with the O_w: after a fixed number of sensor measurements. GRU based architectures outperform LSTM ones.

polishing an object. The dataset was generated with an ATI Multi-Axis Force / Torque Sensor fastened to the wrist of the robot. The human-robot interaction setup is shown in Fig. 1. In the industrial cooperative task the robot is in charge of the picking and placing tasks, while the operator can inspect and polish the object. While the robot offers the object to the operator, three possible intentions are considered: *polish* the object, *grab* the object for a further inspection and *move* the robot to a different pose. Sensor measurements were taken at a frequency of 500 Hz. Each sample consists in a 6 dimensional input: one for each force/torque axis. Samples range from half a second to three seconds. Fig. 4 shows a sample sequence for each of the 3 actions.

The original dataset contains two subsets, one conformed with samples recorded with standardized actions and another one with natural actions. The natural dataset is the one used in this work since it contains more ambiguity among samples of different classes.

Experiments consists in a cross-validation without replacement applied 10 times, data is randomly split between training (75 %) and test (25 %) sets. The performance is evaluated in terms of F1-score. Results are evaluated at different window's sizes (t_i): 0.1, 0.2, 0.5, 0.7 and 1s. As we strive for an early detection of the intentions, inference time for each of the methods is also considered.

B. Network's architecture evaluation

The proposed network first is evaluated with the O_w criterion in order to have an evaluation at the same time instants as [4]. Fig. 5 shows F1-measure results of RNN with this approach. Results of the proposed architectures are over 0.85 even when a small number of hidden states is used. Notice that GRU based methods achieve better results than their LSTM counterparts. As expected, the performance increases when using more hidden units. In the results, it stands out the performance achieved with few sensor measurements (0.1 and 0.2 seconds windows). The best performing method, GRU 200 window, obtains 0.937 in F1-measure at the 0.1 seconds window.

More complex configurations with higher number of hidden states and extra layers were tested without yielding better results than the ones presented in this section. The results of this phase indicate that 200 hidden units is the amount of complexity needed to represent the complexity of the dataset. During the following evaluations, only results with 200 hidden units are shown.

C. Window-Based Evaluation

In this section we compare the the best configuration for our method under the O_w criterion (see Section IV-B) using configurations with 200 hidden units against the best performing methods GPLVM and DTWi in [4]. The results are displayed in Fig. 6 where we can observe that O_w based methods obtain higher F1-scores than GPLVM and DTWi when shorter windows are analyzed. Those situations are the ones targeted in this work, since our aim is to infer the actions performed by the user in the shortest possible time. GPLVM and DTWi methods keep improving their performance when considering longer windows, while the performance of RNN methods saturates. This maximum performance obtained is not a huge limitation of our method, since it occurs at 0.972 F1-score for the GRU 200 configuration.

Another factor to consider when comparing the different methods is the processing time needed to classify the action. For RNN results, CPU and GPU executions are compared. The CPU used is a i7-7800X CPU at 3,5 GHz and the GPU a GeForce GTX 1080 Ti. For both configurations, only one measurement is processed at the time (batch size equal to 1). Times for RNN are provided as the mean time of processing an entire window. GPLVM and DTWi processing times are taken from their original article. Table I shows a comparison of the processing times of the proposed methods compared to GPLVM and DTWi. Notice that for the proposed method, the network processes the sensor measurement while they

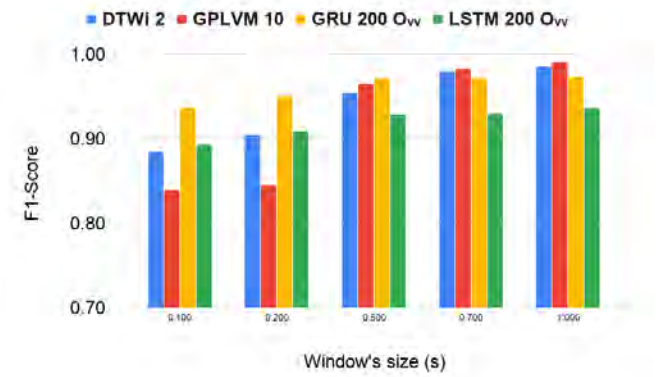

Fig. 6: RNN results compared with window-based techniques. GRU based method outperforms the window-based techniques when using windows smaller than 0.5 seconds.

	0.1 s	**0.2 s**	**0.5 s**	**0.7 s**	**1.0 s**
GPLVM 10	117	130	116	120	135
DTWi 2	8	13	57	108	178
GRU 200 O_w gpu	3	3	5	5	7
GRU 200 O_w cpu	8	7	12	12	17
LSTM 200 O_w gpu	3	3	5	5	7
LSTM 200 O_w cpu	9	7	13	13	18

TABLE I: Time Evaluation. Processing time in milliseconds for O_w criterion compared with window-based techniques. The network proposed for the task processes each sensor measurement, the processing time grows linearly to the number of used sensor measurements. GPU executions are 2.5 times faster than the CPU ones.

are generated. This implies that the time measure provided in Table I for the RNN implementation is higher than the actual introduced latency, which is the time to process the last sample. Nevertheless, results are provided for the full sequence for a fair comparison with the other methods. GPU execution is about 2.5 times faster than the CPU for the LSTM/GRU 200 hidden units. In both cases, times are below 0.02 seconds obtaining much faster response times than GPLVM and DTWi algorithms.

Summarizing, we can conclude that GRU based RNN obtains higher F1-measure and a similar processing time than LSTM. We will use GRU with 200 hidden units for the remainder of the article. In [4], the proposed method is GPLVM 10, which leads to a 0.981 and an inference time of 0.85s (0.7 window + 0.15 processing time). The O_w criterion proposed using a GRU 200 method achieves 0.937 with a processing time of 0.103 seconds. The proposed method provides 8.5x faster decision time at the cost of 4.4 points in F1 measure. Furthermore, the O_w provides a classification for each sensor measurement, while DTW and GPLVM work with windows of the sensor measurements. Our approach greatly reduces the robot's response time while maintaining a F1-measure over 0.90.

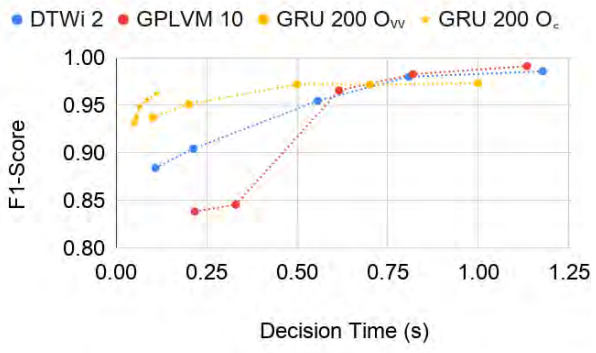

Fig. 7: RNN results with O_C confidence based decision. O_c confidence based criterion (GRU 200 confidence) reduce the mean inference time of getting a decision while increasing the F1-measure compared to O_w (GRU 200 window).

		Predicted label											
		0.4			0.6			0.8			0.95		
		G	M	P	G	M	P	G	M	P	G	M	P
True label	G	95	4	1	96	4	0	97	3	0	97	3	0
	M	7	87	6	6	88	6	6	89	5	4	92	4
	P	1	4	95	1	3	96	1	2	97	1	1	98

TABLE II: Confusion Matrices for O_c criterion with **G**rab, **M**ove and **P**olish intentions. Multiple values for the confidence threshold th_c (0.4, 0.6, 0.8 and 0.95) are used, leading to higher F1-measure results.

D. Confidence-Based Evaluation

In this section we compare the O_w criterion results from Section IV-C with the O_c criterion, both with the GRU 200 configuration. For this experiment, we explore different confidence thresholds th_c : 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 0.95. We expect the less restrictive threshold will promote quicker decisions while the most restrictive will obtain higher F1-measure after analyzing more sensor measurements.

In Section IV-C we concluded that a window of 0.1 seconds is enough for O_w criterion to achieve a satisfactory accuracy for the studied industrial application. For the O_c criterion, we fixed the t_{c_0} parameter (number of samples to update the hidden states) empirically to half this value: 0.05 seconds.

Results in this section are provided comparing the decision time taken for each method. The decision time is obtained summing up the window time and the time needed for the algorithm to make the decision. Thus, we compute the time elapsed between the first sample is generated and when the decision is taken.

Fig. 7 shows results for the GRU 200 method with the O_w and the O_c criteria. As O_c does not have a fixed window to do the decision, time results are provided as the mean time needed to do the decision. With the GRU 200 O_c and a decision threshold of 0.95, a 0.962 F1-score is achieved with an average window size of 0.112 seconds. Comparing with the window-based (see Section IV-C), the O_c criterion obtains a slightly higher decision time but it increases 2.5

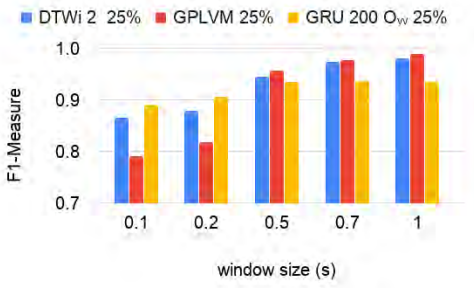

Fig. 8: F1-measures comparison with a reduced training size. Results of GRU 200 and GPVLM with 25% training size are analogous than training with a larger number of samples.

points the F1-measure. Comparing with the GPVLM 10, the O_c criterion provides 7.6x faster decision at the cost of 1.9 points in F1 measure.

A deeper understanding of the net is provided thought the analysis of the confusion matrices for the O_c criterion in Table II. *Move* is the hardest of the three actions since it's erroneously classified as *grab* or *polish*. Setting a more restrictive th_c increases the percentage of correctly detected *move* samples from 87% to 92%. There is almost no confusion between *polish* and *grab* intentions. Examining carefully the dataset we found that the *move* action is the one that has higher variation. Since the operator can move the robot towards any direction and with different forces, this action is the hardest to classify.

E. Limited training data

Finally, we want to analyze one of the limitations of deep learning techniques which is the need of extensive training data. We compare the O_w criterion, the GPVLM and the DTWi using a reduced training set's size of 25%. Results are shown in Fig. 8. We can observe that for shorter window sizes GRU still outperforms the other two methods. Comparing the F1 score obtained using a training set of 75% (see Fig. 6) we can observe that, as expected, the use of a reduced training set affects all the 3 methods. Notably, the F1-score of the GRU method is of almost 90% for the shorter windows whereas for longer windows DTWi and GPLVM take advantage of having more information.

V. Conclusions

We proposed and validated a method to detect user intentions in a human robot collaborative application. Our method takes advantage of the sequential nature of the force sensor data analyzed by using a recurrent neural network that detects the interaction with a confidence value for each sensor measurement. We first defined a time-based criterion (O_w) to demonstrate that the network obtains higher F1-measure than other methods in the literature that use a window approach. We additionally presented a confidence-base criterion (O_c) in order to generate a continuous real-time decision. The continuous nature of the O_c classification closes the loop in real time, introducing a trade-off between speed and accuracy. The method did not suffer when trained with limited data, enabling a quick deploy in new applications without many setup involved. As a future work, RNN can be extended to work with multiple sensors data, whether it is more force sensors, 3D localization sensors, information coming from cameras or from voice commands.

Acknowledgment

This work is supported by the Regional Catalan Agency ACCIÓ through the RIS3CAT2016 project SIMBIOTS (COMRDI16-1-0017), the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI (Institut de Robòtica i Informàtica Industrial) (MDM-2016-0656), the HuMoUR project TIN2017-90086-R (AEI/FEDER, UE), and the European Social Fund and the Ministry of Business and Knowledge of Catalonia through the FI 2020 grant.

References

[1] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design considerations for safe human-robot collaborative workplaces," *Procedia CIRP*, vol. 37, pp. 248–253, 12 2015.

[2] A. Rojko, "Industry 4.0 concept: background and overview," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 5, pp. 77–90, 2017.

[3] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248 – 266, 2018.

[4] A. Olivares-Alarcos, S. Foix, and G. Alenyà, "On inferring intentions in shared tasks for industrial collaborative robots," *Electronics*, vol. 8, no. 11, p. 1306, 2019.

[5] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *CoRR*, vol. abs/1801.01078, 2018.

[6] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A Review of Intent Detection, Arbitration, and Communication Aspects of Shared Control for Physical Human–Robot Interaction," *Applied Mechanics Reviews*, vol. 70, 02 2018. 010804.

[7] P. Maurice, V. Padois, Y. Measson, and P. Bidaud, "Human-oriented design of collaborative robots," *International Journal of Industrial Ergonomics*, vol. 57, pp. 88 – 102, 2017.

[8] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous Robots*, 10 2017.

[9] A. Huentemann, E. V. Poorten, and E. Demeester, "Estimating powered wheelchair driver intentions more accurately using force feedback information," in *ISR 2018; 50th International Symposium on Robotics*, pp. 1–4, June 2018.

[10] C. Gaz, E. Magrini, and A. D. Luca, "A model-based residual approach for human-robot collaboration during manual polishing operations," *Mechatronics*, vol. 55, pp. 234 – 247, 2018.

[11] Z. Liu and J. Hao, "Intention recognition in physical human-robot interaction based on radial basis function neural network," *Journal of Robotics*, vol. 2019, pp. 1–8, 04 2019.

[12] A.-N. Sharkawy, P. Koustoumpardis, and N. Aspragathos, "Variable admittance control for human-robot collaboration based on online neural network training," 10 2018.

[13] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robotics and Automation Letters*, vol. 4, pp. 740–746, April 2019.

[14] T. Zhou and J. P. Wachs, "Early prediction for physical human robot collaboration in the operating room," *CoRR*, vol. abs/1709.09269, 2017.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[16] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.