Anomaly Detection for Dynamic Human-Robot Assembly

Application of an LSTM-based autoencoder to interpret uncertain human behavior in HRC

Fabian Schirmer*
Institute Digital Engineering,
Technical University of Applied Sciences,
Würzburg-Schweinfurt, Germany
fabian.schirmer@thws.de

Philipp Kranz
Institute Digital Engineering,
Technical University of Applied Sciences,
Würzburg-Schweinfurt, Germany
philipp.kranz@thws.de

Jan Schmitt
Institute Digital Engineering,
Technical University of Applied Sciences,
Würzburg-Schweinfurt, Germany
jan.schmitt@thws.de

Tobias Kaupp
Institute Digital Engineering,
Technical University of Applied Sciences,
Würzburg-Schweinfurt, Germany
tobias.kaupp@thws.de

ABSTRACT

Human-Robot Collaboration (HRC) requires humans and robots to work on the same product in the same work environment at the same time. Therefore, the robotic system needs to understand human behavior so it can assist the human appropriately. Since the human is an uncertain variable in this system, human action recognition is one of the key challenges when it comes to HRC. To address this problem, we developed an anomaly detection framework for the dynamic assembly of complex products. We used an Long-Short-Term-Memory (LSTM)-based autoencoder to detect anomalies in human behavior and post-process the output to categorize it as a green or red anomaly. A green anomaly represents a deviation from the intended order but a valid assembly sequence. A red anomaly represents an invalid sequence. In both cases, the worker is guided to complete the assembly process. We demonstrate our proposed framework using an appropriate industrial use case.

CCS CONCEPTS

• **Computer systems organization** → *Robotic components.*

KEYWORDS

Human robot collaboration, human action recognition, LSTM-based autoencoder, dynamic assembly sequence

ACM Reference Format:
Fabian Schirmer, Philipp Kranz, Jan Schmitt, and Tobias Kaupp. 2023. Anomaly Detection for Dynamic Human-Robot Assembly: Application of an LSTM-based autoencoder to interpret uncertain human behavior in HRC. In *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23 Companion), March 13–16, 2023, Stockholm, Sweden.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3568294.3580100

HRI '23 Companion, March 13–16, 2023, Stockholm, Sweden

ACM ISBN 978-1-4503-9970-8/23/03…$15.00
https://doi.org/10.1145/3568294.3580100

1 INTRODUCTION

In industrial environments several trends are evolving. Firstly, a shift from mass production to the assembly of products with a high variance is recognizable [13]. Secondly, the use of cobots tremendously increases [17]. Simultaneously, the amount of skilled workers decrease according to the demographic change [3]. To address these upcoming trends, HRC combines the strengths of humans and robots. The robot takes on actions that require repetitive accuracy and heavy weights, while the human takes on tasks that require dexterity and logical interpretation. Nevertheless, the factor human is a non-deterministic variable, especially when the process is dynamic. In the context of this late breaking report we define a "dynamic assembly process" as a process that has multiple valid ways to assemble a product.

In this late breaking report, we present a framework to interpret human actions during the assembly process by using an anomaly detection algorithm. In addition, we post-process detected anomalies using an Assembly Sequence Plan (ASP) with multiple possible assembly paths and object detection to classify the anomaly as green or red. A green anomaly is classified if the worker chooses an alternative, but valid way to assemble the product. Since there is a deviation from the intended order an anomaly will be detected by the anomaly detection algorithm, but classified as a valid assembly sequence based on the ASP. A red anomaly is detected when an invalid assembly path is selected. A prerequisite is that the ASP contains multiple options to assemble the product. Our contributions in this late breaking report are:

(1) An anomaly detection framework for worker actions in a dynamic assembly sequence.
(2) A representation of an ASP as a directed graph containing multiple valid paths.
(3) Preliminary results from an LSTM-based autoencoder trained on one valid assembly path.

2 RELATED WORK

This work lies at the intersection of two fields: a) human action recognition, and b) anomaly detection via LSTM-based autoencoders. Both topics are addressed in the following sections.

2.1 Human Action Recognition

Human Action Recognition (HAR) is important for our work since we want to adapt the assembly sequence at run-time to the actions of the human. According to Li et al. [7], bi-directional empathy can be achieved by recognizing human actions and predict their intentions. With this knowledge, the robot can react to uncertainties in the human behavior, which is one of the key challenges when it comes to HRC.

According to Pareek et al. [11] the process of HAR consist of three different steps: 1) action representation, 2) dimensionality reduction and 3) action analysis. Action representation, especially in the context of HRC, is often done by extracting skeleton information using depth sensors like the Kinect [5]. Compared to RGB images and videos, skeleton information primarily consists of one-dimensional information which makes action recognition simpler and more effective [11]. In the presented framework we also use skeleton-based pose estimation as input features for our model. Since we use an LSTM autoencoder, which cannot extract spatial information [6], we only use 2D-pose estimation. An LSTM-based autoencoder is able to address both dimensionality reduction and action analysis [16]. The high dimensional features are transformed to low level dimensions with minor errors and the LSTM is able to cover temporal dependencies within the actions.

2.2 LSTM-Based Autoencoder

In our work, we frame HAR as the detection of anomalies. According to Sodemann et al. [14] there are three common types of anomalies in research: 1) anomalous events with significantly different characteristics from normal events (e.g. spatial, color); 2) anomalous events with temporal irregularities (e.g. happen rarely, wrong order); 3) anomalous events that have a specific meaning. Soti et al. [15] look at the first two types of anomalies in the context of HRC. They use an LSTM to detect "temporal normal behavior" of the human that can change over time. The algorithm is able to react to the uncertainties in human behavior. Our algorithm has the same goal, but we extend the focus to the third category of anomalies by post-processing the detected anomalies and compare them with assembly sequences and object related information and hence give them a meaning.

Provotar et al. [12] state, that classical machine learning (ML) methods (support vector machines, hidden markov models, nearest neighbors) are not suitable when it comes to anomaly detection in time series data. They therefore train an LSTM-based autoencoder on two timeseries datasets. Their model works on all types of time series data but is likely to have high false positive rates (38 %). With our novel framework we are able to reduce the false positive rate of LSTM-based autoencoders by post-processing the results with additional object detection and assembly sequence plan information as well as human cognition.

Nassif et al. [10] indicate that deep learning approaches often fail to provide accurate results when it comes to HAR in HRC. They state that limited data from operator assembly tasks are available and that these data suffer huge distribution discrepancy caused by different working conditions and human body characteristics. In our initial work, we limit ourselves to a single assembly sequence and were able to train a robust model on this workflow. Our model only requires limited data since we exclusively train the intended assembly sequence. Alternative sequences (both valid and invalid) are detected as anomalies (green and red). The model can be improved by incorporating feedback from the human operator during the assembly.

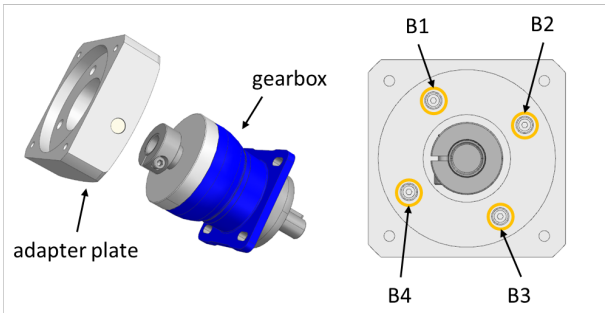

Figure 1: Exploded view of the gearbox assembly on the left and the numbering of the bolts on the right.

3 ANOMALY DETECTION IN DYNAMIC HUMAN-ROBOT ASSEMBLY

An experimental setup based on an industrial use case was used to evaluate our anomaly detection framework and also to explain the term of dynamic human-robot assembly.

3.1 Experimental Setup

The industrial use case to evaluate our approach is shown in Figure 1. An adapter plate is bolted to a gearbox using four bolts.

Table 1 shows the intended assembly sequence. The robot manipulates the heavy parts in step 1 and 2 and holds the adapter plate in position in subsequent steps ("third hand"). The human handles the small components that require a certain dexterity. For our studies we only focus on the bolting part, assuming that the gearbox (GB) and the adapter plate (AP) are already at their final position.

Table 1: Assembly Sequence Plan (ASP) of the final product.

Step	Assembly Description	Assignment
1	Place Gearbox (GB)	Robot
2	Place adapter plate (AP)	Robot
3	Bolt down B1	Human
4	Bolt down B3	Human
5	Bolt down B2	Human
6	Bolt down B4	Human

3.2 Dynamic Human-Robot Assembly

Figure 2 shows the valid assembly paths of our industrial use case. Step 1, placement of the gearbox (GB) and step 2, placement of the adapter plate (AP) have a fixed order. Steps 3-6 can be executed in different orders which we refer to as "dynamic" human robot assembly. The worker can start with any of the four bolts in step 3,

but is then constrained in step 4 because bolts must be mounted diagonally to avoid jamming. In step 5, the worker is again free to choose, but restricted in step 6 since only one bolt is left. In total, there are eight different ways to mount the gearbox correctly. The path highlighted in orange is the one intended for the assembly of the final product (FP). Only this path was used for training the anomaly detection algorithm. Training with all valid assembly paths would be time consuming and would result in a more complex model. The path highlighted in green is a valid alternative path. This path is classified as an anomaly since it was not used for training. The post-processing of our algorithm classifies this path as a valid way to assemble the final product and marks it as a *green anomaly*. The path marked in red would lead to an invalid assembly, since it can cause jamming in the connection between the gearbox and the adapter plate. Our approach recognizes this case as a *red anomaly*. In this case, both the robot and the human must be informed that something went wrong.

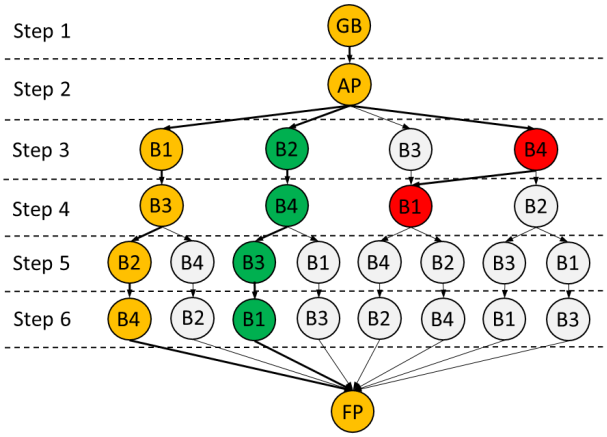

Figure 2: Possible assembly paths in our dynamic assembly process with the intended path highlighted in orange, a green anomaly (valid alternative assembly sequence) and a red anomaly (invalid assembly sequence).

4 SYSTEM ARCHITECTURE

The architecture of our proposed framework consists of four parts as shown in Figure 3: *Input, Interpretation, Post-processing and Output*. The data flow is from left to right. The input data consists of data points from a 2D-pose estimation algorithm. The subsequent part based on an LSTM autoenocder interprets the data based on the intended assembly sequence (orange path). The final post-processing part analyses the anomaly and categorizes it as a *green* or *red anomaly*. The last part is the output unit which gives feedback to the worker as well as guidance for the next steps in the ASP.

4.1 Input

The features to train our algorithm are defined and extracted in the *Input* part. Pareek [11] referred to this as the first step in HAR, "action representation". Single data points are generated by a pose estimation algorithm that marks and extracts joint positions of the

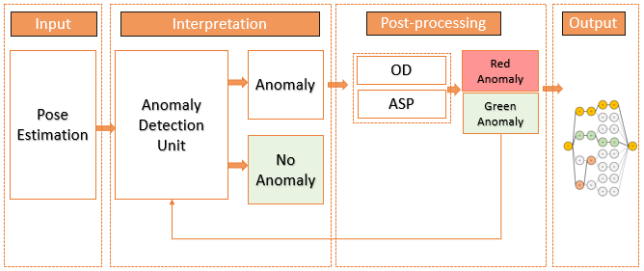

Figure 3: Anomaly detection framework to interpret the assembly process. If an anomaly is detected post-processing will be activated. Object detection (OD) is then used to match the current assembly step with the possible assembly paths in the Assembly Sequence Plan (ASP).

human body out of 2D images [2] [9]. The output are 250 feature points on a single frame. The input of our LSTM autoencoder is a three dimensional array. The first parameter x_1 stands for the number of samples used for training. The second parameter x_2 represents the recorded frames (time steps), and the third parameter x_3 shows the attached feature values. Since the execution time of an assembly step can vary between workers, parameter x_2 does not have a fixed size. For our use case, the variation is small since the human tasks (attaching bolts) are identical for all steps. The third parameter x_3 corresponds to the data points of the joints which are extracted from the pose estimation algorithm. The number of feature points (fixed size of 250) stays the same even if the worker changes.

4.2 Interpretation

In the *Interpretation* unit we combine the HAR steps dimensionality reduction and action analysis by using an LSTM-based autoencoder [8]. The structure of the LSTM autoencoder consists of two parts: 1) the encoder, which is responsible for compressing the input data to a lower dimension, and 2) the decoder, which reconstructs the compressed data to the original input data. The reconstruction error is then used as a threshold to detect anomalies. To interpret the time series of a complete assembly step, we additionally use an LSTM model in order to focus on features in several frames. Thus, our approach is able to detect deviations from the intended order of the ASP by using a threshold for the error value of the algorithm and classifying each step performed by the worker as an anomaly or no anomaly (Figure 4).

4.3 Post-processing

The *Post-processing* unit adds meaning to the recognized anomaly by distinguishing between *green* and *red anomalies*. The following units are used to differentiate between the two: 1) Object Detection (OD) verifies a valid sub-assembly, and 2) Assembly Sequence Plan (ASP) checks whether the alternative way is valid.

The OD unit uses information from an additional camera to detect and locate the objects in the workspace, using the LINE-MOD algorithm [4]. Alternatives such as YOLO or other ML-algorithms

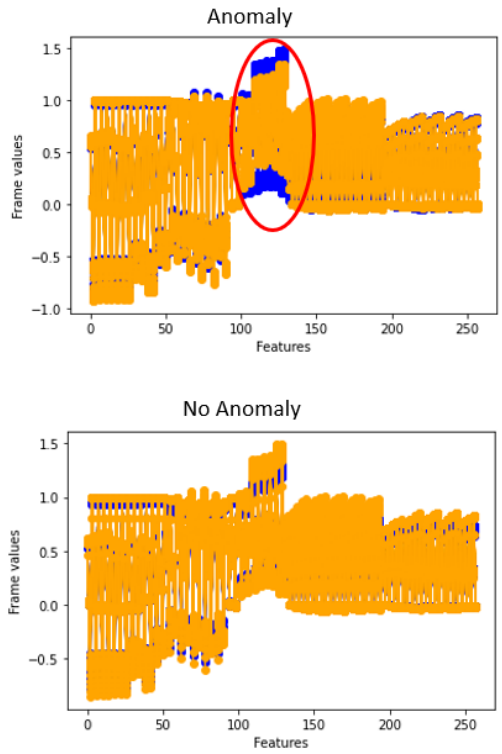

Figure 4: Detected anomaly in the first picture based on the difference between the intended (orange) and an anomaly path (blue). On the second picture no anomaly is detected, because the reconstruction error is below the threshold.

may be used in further work [1, 18]. The localization of the object is used to check whether a valid sub-assembly was assembled within an assembly step and hence whether one of the pre-planned assembly paths is being executed.

Secondly, the ASP unit checks if the assembly step done by the worker is allowed according to the directed graph of assembly sequence plans (Figure 2). Moreover, not only the paths of valid assembly sequences but also the order is examined.

4.4 Output

The *Output* unit guides the worker through the assembly and is used as an interface to input worker feedback to improve the LSTM autoencoder. The interaction with the worker is realised using a worker guidance system (WGS). Three output cases are distinguished:

(1) Intended (orange) assembly path: the WGS displays assembly instructions.
(2) Red anomaly: the WGS flags a possible error to the worker. In case of a false positive, the worker can overrule the system. This information is stored to retrain the algorithm.
(3) Green anomaly: the WGS informs the worker that a non-intended but valid assembly path has been chosen. The information is also stored to improve the algorithm.

5 PRELIMINARY RESULTS

Our experiments show promising results in detecting anomalies with the LSTM autoencoder (Figure 4). For our first evaluation of the training process we used 10 recordings of assembly step 3 and 4, with 30 frames per step. The worker stays the same during each recording. Additionally we tested our models on different sample data (n=30). The results are shown in the confusion matrices of Figure 5.

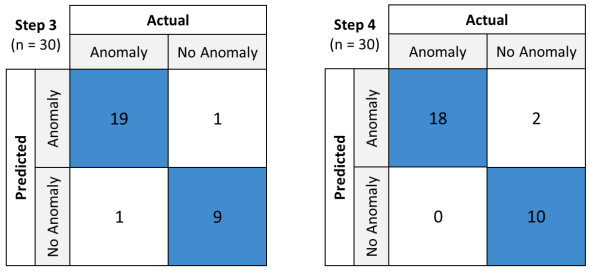

Figure 5: Confusion matrices of step 3 (bolt down B1) on the left and step 4 (bolt down B3) on the right

For the post-processing we tested our object localization algorithm to detect single components and sub-assemblies of our product and matched it with the appropriate assembly sequence step of the ASP-graph. First results revealed that we are able to differentiate between *green* and *red anomalies*.

An overview of the preliminary results is shown in a complementary video attachment [1].

6 CONCLUSION AND FUTURE WORK

In this late breaking report, we have presented a framework on how to detect anomalous human behavior during collaborative human-robot assembly. We use joint positions of the human body, extracted via 2D-pose estimation, as input for an LSTM-based anomaly detection. Using object detection and a directed graph of ASP, we are able to distinguish between valid alternative assembly paths (*green anomalies*) and invalid assembly paths (*red anomalies*). For all cases the worker receives visual instructions on how to proceed and the possibility to correct false positive results via a WGS. In future work, we will use the workers' feedback to improve our algorithm in order to increase the robustness. Furthermore we enhance the post-processing by testing other machine learning detection and localization models and test the framework on a more advanced industrial use case.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the state of Bavaria, Germany. The Bayerische Forschungsstiftung funds the research project KoPro under the grant no. AZ-1512-21. In addition, we appreciate the perspective from our industry partners Fresenius Medical Care, Wittenstein SE, Uhlmann und Zacher, DE software & control and Universal Robots.

[1]This paper has supplementary downloadable material provided by the authors. This includes showing the presented use case with the anomaly detection for a dynamic assembly. This material is 25.1 MB in size.

REFERENCES

[1] Yannick Bukschat and Marcus Vetter. 2020. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach. *arXiv preprint arXiv:2011.04307* (2020).

[2] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. 2013. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3041–3048.

[3] Beate Henschel, Carsten Pohl, and Marcel P Thum. 2008. Demographic change and regional labour markets: The case of Eastern Germany. (2008).

[4] Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. 2011. Gradient response maps for real-time detection of textureless objects. *IEEE transactions on pattern analysis and machine intelligence* 34, 5 (2011), 876–888.

[5] Yanli Ji, Yang Yang, Fumin Shen, Heng Tao Shen, and Xuelong Li. 2019. A survey of human action analysis in HRI applications. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 7 (2019), 2114–2128.

[6] Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. 2017. Skeleton-based action recognition using LSTM and CNN. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 585–590.

[7] Shufei Li, Ruobing Wang, Pai Zheng, and Lihui Wang. 2021. Towards Proactive Human-Robot Collaboration: A Foreseeable Cognitive Manufacturing Paradigm. *Journal of Manufacturing Systems* 60 (July 2021), 547–552. https://doi.org/10.1016/j.jmsy.2021.07.017

[8] Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, and Michael Weyrich. 2021. A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry* 131 (2021), 103498.

[9] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A Framework for Building Perception Pipelines. *CoRR* abs/1906.08172 (2019). http://arxiv.org/abs/1906.08172

[10] Ali Bou Nassif, Manar Abu Talib, Qassim Nasir, and Fatima Mohamad Dakalbab. 2021. Machine learning for anomaly detection: A systematic review. *Ieee Access* 9 (2021), 78658–78700.

[11] Preksha Pareek and Ankit Thakkar. 2021. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review* 54, 3 (2021), 2259–2322.

[12] Oleksandr I Provotar, Yaroslav M Linder, and Maksym M Veres. 2019. Unsupervised anomaly detection in time series using lstm-based autoencoders. In *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. IEEE, 513–517.

[13] Jan Schmitt, Andreas Hillenbrand, Philipp Kranz, and Tobias Kaupp. 2021. Assisted human-robot-interaction for industrial assembly: Application of spatial augmented reality (sar) for collaborative assembly tasks. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 52–56.

[14] Angela A Sodemann, Matthew P Ross, and Brett J Borghetti. 2012. A review of anomaly detection in automated surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 1257–1272.

[15] Gergely Sóti, Ilshat Mamaev, and Björn Hein. 2020. A Modular Deep Learning Architecture for Anomaly Detection in HRI. In *International Conference on Interactive Collaborative Robotics*. Springer, 295–307.

[16] Amin Ullah, Khan Muhammad, Ijaz Ul Haq, and Sung Wook Baik. 2019. Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments. *Future Generation Computer Systems* 96 (2019), 386–397.

[17] Darrell M West. 2018. *The future of work: Robots, AI, and automation*. Brookings Institution Press.

[18] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. 2022. RNNPose: Recurrent 6-DoF Object Pose Refinement with Robust Correspondence Field Estimation and Pose Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14880–14890.