

# HMM-based anomaly interpretation for intelligent robots in Industry 4.0

Alberto Castellini<sup>1</sup>, Cristian Morasso<sup>1</sup> and Alessandro Farinelli<sup>1</sup>

<sup>1</sup>University of Verona, Department of Computer Science, Strada Le Grazie 15, 37134, Verona, Italy

## Abstract

We apply an anomaly detection method based on Hidden Markov Models and Hellinger distance to a Kairos mobile robot operating in the ICE lab, a research laboratory for Industry 4.0. Two main contributions are proposed: *i*) a decomposition of the Hellinger distance which allows to identify the causes of anomalous behaviours detected, *ii*) a graphical user interface that synchronously shows the robot movements in a map and the evolution of the Hellinger distance components, allowing a quick investigation of the causes of the detected anomalies. The tools are applied to a real-world dataset allowing to discover that an anomalous movement of the Kairos robot is caused by a wrong reading of the lidar from a window in the environment.

## Keywords

Anomaly detection, mobile robot, Kairos, Industry 4.0, Interpretability, Explainability

## 1. Introduction

A goal of Artificial Intelligence (AI) is to allow robots to leave controlled environments and take part in our everyday life. A key aspect of long term autonomy [1] to reach this goal is safety. A robot should preserve itself, the environment and humans moving in it, while acting to reach a goal. Important tools for achieving safety in robot autonomy are anomaly detectors [2, 3, 4] able to recognize infrequent data samples possibly related to anomalous situations faced by the robot, or damages in robot's equipment.

In this work we apply a recent online anomaly detection technique [5] based on Hidden Markov Models [6] to data acquired by sensors and actuators of a mobile robot, namely, an RB-Kairos<sup>1</sup>, acting in a real environment for industrial research. The environment is the Industrial Computer Engineering (ICE) lab<sup>2</sup> of the Verona University (Italy), a laboratory for Industry 4.0 with a modern production line, extended with equipment for augmented reality and digital production. The robot has to move between two positions in the laboratory and abnormal behaviours are observed. The anomaly detection technique efficiently learns nominal behaviours of the robot from sequences of sensor readings and recognizes anomalous behaviours instead of single anomalous observations. Then, using the Hellinger distance [7] between the distribution


---

*The 9th Italian Workshop on Artificial Intelligence and Robotics – AIRO 2022*

✉ alberto.castellini@univr.it (A. Castellini); cristian.morasso@studenti.univr.it (C. Morasso);  
alessandro.farinelli@univr.it (A. Farinelli)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup><https://robotnik.eu/products/mobile-manipulators/rb-kairos/>

<sup>2</sup><https://www.icelab.di.univr.it/>

of data observed online (in a time window) and the most similar distribution in the learnt model of nominal behaviour, the detector successfully detects anomalies in the data.

Besides successfully applying the online anomaly detector to a real-world domain, in this work we propose two other contributions. From a methodological point of view we introduce a decomposition of the Hellinger distance in elements focused on specific variables (i.e., sensors and actuators). This decomposition supports the identification of the causes of the anomalous behaviours. From an application viewpoint we provide a prototypical Python GUI that synchronizes a *map view* of the robot moving in the environment with a *chart view* showing the time evolution of the components of the Hellinger distance, for all variables (i.e., sensors or actuators). This GUI allows an informative analysis of the causes of observed anomalies. In the results section we analyze a real anomaly detected by the system and provide interpretation using the proposed tools.

## 2. Related work

Some works employ deep neural networks for online anomaly detection in robot systems, three recent examples are [8, 9, 10], which employ LSTM variational autoencoders. However, these works cannot be considered as alternatives to our online method since they require datasets composed of thousands of execution traces sampled at high frequency. Unlike neural network based techniques, our methodology is sample efficient, in fact in our case study the model of the nominal behavior is generated using only two trajectories travelled by the robot in the ICE environment.

Other key differences are present between our approach and state-of-the-art methods for anomaly detection in autonomous robots. For instance, some works use supervised machine learning approaches to classify data produced in real-time by a robot. The problem of these methods is that they need fully labelled data, which are rarely available in the the context of anomaly detection. Instead our method is completely unsupervised [2], hence it allows to detect even completely unknown anomalies, since it considers anomalous the behaviours that have been never (or rarely) seen in nominal dataset. In [11] an online multivariate data-driven fault detection approach is presented using the Mahalanobis-distance to compare correlated streams of data with previously observed data. In [12], a self-awareness approach is proposed which builds a probabilistic model on the basis of the whole discrete event-based data interchange inside the robot. Finally, some works [13, 11] explicitly deal with contextual faults [2]. The difference with our approach is that we represent different contexts as different states of an HMM, while those in the literature consider recent past observations as context.

Among the approaches in the literature using HMM for anomaly detection, the ones most similar to our approach are [14] and [15], in which HMMs are trained using multimodal sensory signals for detecting anomalies in assistive robots. At run time, the trained HMMs provide likelihood scores for data inside a window, which are compared to an adaptive detection threshold to identify putative anomalies. The methodology that we use substitutes the likelihood estimation with the computation of a more informative and interpretable measure, i.e., the Hellinger distance, that in this work we decompose allowing the identification of the specific sensors involved in the anomaly. A few other recent works aim to improve the interpretability of

the anomaly detection using hybrid data-driven/symbolic approaches. Some of them [16, 17] use of Satisfiability Modulo Theory (SMT) to instantiate pre-defined logic rules designed using prior knowledge. Others [18] use Inductive Learning of Answer Set Programs (ILASP) to automatically detect behavioral patterns in paradigmatic simulation tasks performed by autonomous agents operating in uncertain environments. These methods, however, mainly aim to describe in a human-understandable way (i.e., using logic representations) the behaviours of the robot, while the approach presented in this paper aims to explain the source of observed anomalies using a probabilistic and only data driven approach.

### 3. Dataset

The dataset has been collected by sensors installed on a RB Kairos moving in the ICE lab environment. The environment, shown in the left-hand side of Figure 1, contains a reconfigurable production line with a storage system, from which the robot takes the raw materials and puts the final products, and several tools (e.g., manipulators, a subtractive CNC machine, a 3D printer) used to generate and assemble the product. In the specific data acquisition sessions the robot had to move from the starting point to the ending point in the map. Three testing sessions were performed: two of them had nominal behaviours (see the red and blue trajectories in the map of Figure 1) and the third session shown an anomalous behaviour (see the green line in the map which moves close to a window instead of towards the end of the path).

The raw dataset has about 2300 variables and different number of time steps for different sensors, since each sensor has (potentially) different sampling interval. We focused our analysis on robot position (i.e., coordinates  $X$  and  $Y$ ), robot orientation ( $O$ ) and distances between the robot and surrounding environment/objects. Distances are perceived by lasers. The robot collects 540 laser signals, with sampling angle of  $0.5^\circ$ . We performed data pre-processing with two main goals: to standardize the sampling interval and to reduce the resolution of the lasers. The first operation was performed because the anomaly detection technique requires an uniform sampling interval; the second operation was performed to reduce the dimensionality of the problem, since high dimensionality may generate mathematical issues in model learning and in the computation of the Hellinger distance. In particular, we generated three laser variables, that we called, Laser Left ( $LL$ ), Laser Center ( $LC$ ) and Laser Right ( $LR$ ).  $LL$  is the average of laser signals with angles between  $0^\circ$  and  $89.5^\circ$ , which scan the environment on the left of the robot.  $LC$  is the average of laser signals with angles between  $90^\circ$  and  $179.5^\circ$ , which scan the environment in front of the robot.  $LR$  is the average of laser signals with angles between  $180^\circ$  and  $270^\circ$ , which scan the environment on the right of the robot.

After the pre-processing phase we obtained two datasets, each with six variables (i.e., columns), namely,  $X$ ,  $Y$ ,  $O$ ,  $LL$ ,  $LC$  and  $LR$ . The training dataset contains two nominal trajectories of 32 seconds each, for a total of 64 seconds and 340 time steps (i.e., rows), using a time interval of 0.2 sec. The test set contains a trajectory of 45 seconds (230 time steps) which is nominal for 11 seconds and then it diverges from the nominal path moving towards a window in the environment (see the green line in Figure 1).

## 4. Method

We first describe the main properties of the anomaly detection technique proposed in [5]. Then we introduce the two contributions of this work: the decomposition of the Hellinger distance and the Graphical User Interface. Both support the identification of the causes of the anomalies.

### 4.1. Online anomaly detection technique

Given a general  $d$ -dimensional time series  $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_n\}$  composed of  $n$  observations, where  $\mathbf{o}_t$  is a  $d$ -dimensional vector representing the multivariate (multi-valued) observation at time  $t$ , the nominal behavior (i.e., training set) of the robot is represented as  $\mathbf{O}^N = \{\mathbf{o}_1^N, \dots, \mathbf{o}_{n^N}^N\}$  and the observed behavior (test set) of the same system along some time period as  $\mathbf{O}^O = \{\mathbf{o}_1^O, \dots, \mathbf{o}_{n^O}^O\}$ .

According to [5], the nominal behavior of the robot system is modeled as a HMM  $\lambda^N$  that is learned from the training set  $\mathbf{O}^N$  using the Baum-Welch algorithm [6]. The number of hidden states is selected by minimizing the BIC. Online anomaly detection at time step  $t$  is performed by means of a sliding window  $\mathbf{W}_t = \{\mathbf{o}_{t-w+1}^O, \dots, \mathbf{o}_t^O\}$  of length  $w$  which includes the last  $w$  observations. For each window  $\mathbf{W}_t$ , a score is computed and, when the score exceeds a predefined threshold  $\tau$ , the behavior is considered anomalous. The score is the Hellinger distance between the estimated distribution of the observations corresponding to the most frequently occurring state  $\hat{s}_t$  in the Viterbi path  $S_t = \{s_{t-w+1}, \dots, s_t\}$  of window  $\mathbf{W}_t$  and the emission probability of the same state in  $\lambda^N$ .

The online procedure computes the Viterbi path of the multivariate time series inside the window. For the state  $\hat{s}_t$  occurring most frequently in the Viterbi path a multivariate Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is fit with the data inside the window. Then the Hellinger distance is computed between  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  and the emission probability of state  $\hat{s}_t$  in  $\lambda^N$  using the formula

$$H(f, g) = 1 - \frac{\det(\boldsymbol{\Sigma}_1)^{1/4} \det(\boldsymbol{\Sigma}_2)^{1/4}}{\det\left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)^{1/2}} \cdot \exp\left\{-\frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right\}. \quad (1)$$

If the distance is larger than a threshold  $\tau$ , then a warning is reported. Full details of the procedure are described in [5].

### 4.2. Decomposition of the Hellinger distance

The second term of Equation (1) determines the size of the Hellinger distance for each window, namely, if this term is close to 1 then the Hellinger distance is close to 0, indicating a nominal behaviour, and if the term is close to 0 then the Hellinger distance becomes close to 1, reporting an anomalous behaviour. The second term can be split, again, in two factors. The first one,

$$\frac{\det(\boldsymbol{\Sigma}_1)^{1/4} \det(\boldsymbol{\Sigma}_2)^{1/4}}{\det\left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}\right)^{1/2}}$$

depends on the differences in the covariance matrices  $\Sigma_1$  and  $\Sigma_2$  of the two distributions, while the second factor,

$$\exp\left\{-\frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_1 - \mu_2)\right\}$$

depends on both the means  $\mu_1$  and  $\mu_2$  and the covariance matrices  $\Sigma_1$  and  $\Sigma_2$  of the two distributions. Considering  $n$  variables  $x_1, \dots, x_n$ , with means  $\mu_1 = (\mu_{1,1}, \dots, \mu_{1,n})$  and  $\mu_2 = (\mu_{2,1}, \dots, \mu_{2,n})$  for distributions  $f$  and  $g$ , respectively, and assuming diagonal covariance matrices

$$\Sigma_1 = \begin{pmatrix} \sigma_{1,1}^1 & \dots & 0 \\ 0 & \sigma_{i,i}^1 & 0 \\ 0 & \dots & \sigma_{n,n}^1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} \sigma_{1,1}^2 & \dots & 0 \\ 0 & \sigma_{i,i}^2 & 0 \\ 0 & \dots & \sigma_{n,n}^2 \end{pmatrix}$$

for distributions  $f$  and  $g$ , respectively, we developed the matrix calculations in the formula of the Hellinger distance and grouped by variable obtaining two Hellinger terms for each variable  $i$ . Notice that the assumption of diagonal covariance matrices is reasonable in several applications. It only avoids considering covariances between different variables, hence the tool would not detect anomalies related to different covariances between nominal and observed behaviours. When small amount of data are available, as in our case study, the estimation of covariance parameters is very error prone, hence diagonal covariance matrices are used anyway in these cases that represent the majority of cases in real-world applications. The first term (Eq. (2)) is related to the first factor in the Hellinger formula, and the second term (Eq. (3)) decomposes the exponential in the second factor of the Hellinger formula. For the  $i$ -th variable the two terms are:

$$H_1(i) = \frac{\sqrt{2} \left(\sigma_{i,i}^1 \cdot \sigma_{i,i}^2\right)^{1/4}}{\left(\sigma_{i,i}^1 + \sigma_{i,i}^2\right)^{1/2}} \quad (2) \quad H_2(i) = \frac{(\mu_{1,i} - \mu_{2,i})^2}{4(\sigma_{1,i} + \sigma_{2,i})} \quad (3)$$

According to this decomposition the Hellinger distance can be written as

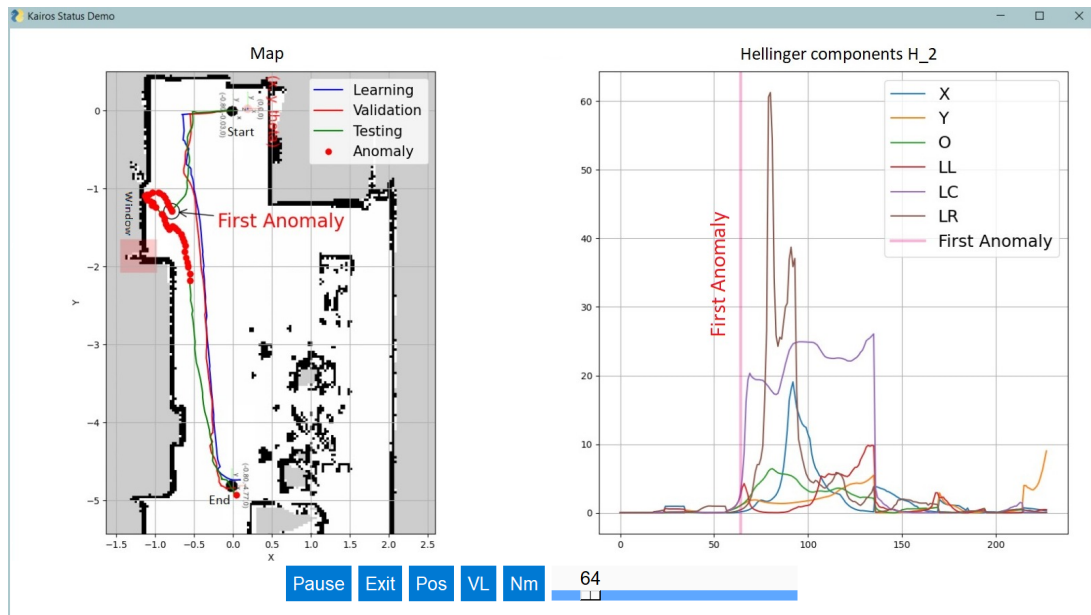
$$H(f, g) = \prod_{i=1 \dots n} H_1(i) \cdot \exp\left\{-\sum_{i=1 \dots n} H_2(i)\right\} \quad (4)$$

Components  $H_1(i)$  and  $H_2(i)$  are very useful to identify the variables more involved in an identified anomalies. In the next section we show a real case study in which this decomposition allowed to identify the causes of an anomalous behaviour of the RB Kairos in the ICE lab.

### 4.3. Graphical user interface

The Python graphical user interface (GUI) here proposed is displayed in Figure 1. It shows the map of the environment on the left and the evolution of Hellinger components  $H_1(i)$  and  $H_2(i)$  on the right. The GUI is updated in real-time as new data are loaded from a csv file. At the beginning the marker representing the robot (a black circle) is positioned in the starting point

of the map and the nominal trajectories (blue and red lines) are shown. In the chart on the right, the pink vertical line showing the time instant is placed on coordinate 0 of the x-axis. As time goes on, the robot marker is moved in the map and the related green trajectory is plotted. On the right, the green vertical line is moved to the right and the chart of Hellinger components for each variable is generated (in Figure 1 the entire chart is shown because the dataset is analyzed offline). In the bottom of the GUI some buttons allow to start/stop the run and a slider allows to quickly move forward and backward if the dataset is analyzed offline. The online version of the GUI only shows the trajectory and the chart until the current time instant and the slider allows only to focus on previously seen parts of the robot trajectory. Other commands allow to switch between different HMM models using different sets of variables (if they were learnt before).



**Figure 1:** Graphical User Interface. Right: map of the environment showing robot movements. Left: Hellinger components showing the contribution of each variable to the Hellinger distance. High values identify the variables more involved in anomalous behaviours.

## 5. Results

We computed HMM models of nominal behaviors from the training set defined in Section 3 and using 4 different combinations of variables. We computed the accuracy of each model when used in the anomaly detector. The model of greatest interest was that using variables  $X$ ,  $Y$ ,  $O$ ,  $LL$ ,  $LC$  and  $LR$ . With parameter tuning we identified a good set of parameters, namely, 9 hidden states in the HMM and a window of 20 steps (i.e., trajectory intervals of about 3 seconds are considered to identify the anomalies). The accuracy of this model is 0.95 (with a precision of 1.00, a recall of 0.88 and a F1-score of 0.94). In the computation of the accuracy we used a ground truth manually defined considering the distance of the trajectory from nominal trajectories.

We tested the anomaly detector on the test set defined in Section 3 containing a known anomalous trajectory. As shown in Figure 1, the detector managed to identify the anomaly (see the red dots on the green trajectory). In particular, if we focus on the point where the anomaly starts (see the circle in the map and the corresponding pink vertical line in the chart) we see that the first Hellinger component that grows is that of variables LL and LC (the left and central lasers, see the red and purple lines). After about 2 seconds, the Hellinger component of variable LR (laser right) grows quickly and reaches a very high level. Only after 2 seconds the Hellinger component of the X position grows (blue line in the chart). This analysis shows that the causes of the anomalous behaviour of the robot, which strangely moves close to the window, are anomalous observations from the laser. For this analysis we only shown components  $H_2(i)$  because components  $H_1(i)$  were not informative (always close to 1) in this case study.

From a discussion with robot operators it emerged that a probable cause of the anomaly was the fact that the window during the testing trajectory was not covered by a curtain as it was during the nominal trajectories, hence the robot probably received wrong distances from the central laser which observed out of the room, it started moving towards the window to explore an area that it did not explored before, and moving closer to the window also the right laser started to get anomalous readings from the window. Hence the anomalous laser readings can be identified as very probable causes of the anomalous trajectory travelled by the robot.

## 6. Conclusions and future work

The tests performed in this work on a RB Kairos robot show that the online anomaly detection methodology investigated is able to identify anomalous behaviours of robot. Furthermore, the decomposition of the Hellinger distance allows to improve the interpretability and to identify the causes of the anomalous behaviours. The GUI supports the analysis both in real-time and offline. Future developments will focus on comparing the performance and the interpretability of this method with those of other anomaly detectors in the literature. Then we want to integrate the detector and the GUI in the ICE-lab demonstrator to show its capability in real-time applications.

## Acknowledgement

The authors would like to thank Francesco Trotti for collecting the data from the Kairos robot and the ICE Lab of the University of Verona for allowing the use of robots. The research has been partially supported by the projects "Dipartimenti di Eccellenza 2018-2022, funded by the Italian Ministry of Education, Universities and Research (MIUR).



## References

- [1] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, T. Krajník, Artificial intelligence for long-term robot autonomy: A survey, *IEEE RA-L* 3 (2018) 4023–4030.
- [2] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM Comput Surv* 41 (2009) 1–58.
- [3] E. Khalastchi, M. Kalech, On fault detection and diagnosis in robotic systems, *ACM Comput Surv* 51 (2018) 9.
- [4] W. Aoudi, M. Iturbe, M. Almgren, Truth will out: Departure-based process-level detection of stealthy attacks on control systems, in: *Proc. CCS*, 2018, p. 817–831.
- [5] D. Azzalini, A. Castellini, M. Luperto, A. Farinelli, F. Amigoni, Hmms for anomaly detection in autonomous robots, in: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '20, IFAAMAS*, 2020, p. 105–113.
- [6] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, *P IEEE* 77 (1989) 257–286.
- [7] E. Hellinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen, *Journal für die reine und angewandte Mathematik* 136 (1909) 210–271.
- [8] D. Azzalini, L. Bonali, F. Amigoni, A minimally supervised approach based on variational autoencoders for anomaly detection in autonomous robots, *IEEE Robotics Autom. Lett.* 6 (2021) 2985–2992. doi:10.1109/LRA.2021.3062597.
- [9] D. Park, Y. Hoshi, C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder, *IEEE RA-L* (2018) 1544–1551.
- [10] M. Sölch, J. Bayer, M. Ludersdorfer, P. van der Smagt, Variational inference for on-line anomaly detection in high-dimensional time series, in: *Proc. ICML 2016 Anomaly Detection Workshop*, 2016.
- [11] E. Khalastchi, M. Kalech, G. Kaminka, R. Lin, Online data-driven anomaly detection in autonomous robots, *Knowl Inf Syst* 43 (2015) 657–688.
- [12] R. Golombek, S. Wrede, M. Hanheide, M. Heckmann, Learning a probabilistic self-awareness model for robotic systems, in: *Proc. IROS*, 2010, pp. 2745–2750.
- [13] M. Kalisch, Fault detection method using context-based approach, in: Z. Kowalczyk (Ed.), *Adv. and Intell. Computations in Diagnosis and Control*, Springer, 2016, pp. 383–395.
- [14] D. Park, Z. Erickson, T. Bhattacharjee, C. Kemp, Multimodal execution monitoring for anomaly detection during robot manipulation, in: *Proc. ICRA*, 2016, pp. 407–414.
- [15] D. Park, H. Kim, C. Kemp, Multimodal anomaly detection for assistive robots, *Auton Robot* 43 (2019) 611–629.
- [16] G. Mazzi, A. Castellini, A. Farinelli, Identification of unexpected decisions in partially observable monte-carlo planning: A rule-based approach, in: *Proc. 20th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS), IFAAMAS*, 2021, p. 889–897.
- [17] G. Mazzi, A. Castellini, A. Farinelli, Rule-based shielding for partially observable monte-carlo planning, in: *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, volume 31, 2021, pp. 243–251.
- [18] G. Mazzi, D. Meli, A. Castellini, A. Farinelli, Learning logic specifications for soft policy guidance in pomcp, in: *Proceedings of the 22nd Conference on Autonomous Agents and MultiAgent Systems*, 2023, p. in publication.