Privacy-Preserving User Pose Prediction for Safe and Efficient Human-Robot Interaction

Adel Baselizadeh[1], Weria Khaksar[2], Md Zia Uddin[3] , Diana Saplacan[4] , and Jim Torresen[5]

***Abstract*— Enhancing user privacy is crucial in improving the safety and efficiency of Human-Robot Interaction (HRI), as it is a key factor for establishing user trust in the robot. Using privacy-preserving sensors and local processing of the user's data are ways to enhance privacy in HRI. This paper presents a privacy-preserving sensing system for real-time tracking and predicting the user's movements in HRI. As privacy-preserving sensors, a thermal and a depth camera are used to monitor the user's movements and determine their current pose. In order to improve the robot's perception of the user's situation and enhance the quality of real-time user monitoring, a Deep Learning (DL) model has been developed to estimate the future poses of the user. The developed model is based on the Sequence to Sequence mechanism (Seq2Seq). Modifications have been made to Seq2Seq so it can be run locally on the robot. As a result of these modifications, the computational cost of the model has been reduced by 34%. Experimental studies have been conducted to evaluate the performance of the sensing system in tracking and predicting the user's movements in HRI. According to the test results, the proposed sensing system is able to track the movements of the user appropriately. Additionally, it is shown that the estimation of the user's movement through the proposed system with the prediction model can improve the safety and efficiency aspects of the HRI experiments by up to 24% and 17%, respectively.**

I. INTRODUCTION

Robot sensing is one of the prevalent characteristics of robot autonomy, amongst sensing, planning, and acting [1]. It plays a significant role in HRI since it enables the robot to understand the user's condition and movements in order to perform more efficiently and safely in their presence. Therefore, it has always been essential to equip robots with sensors that can collect detailed and accurate data about the environment and the user. Additionally, proper processing of the data collected by the sensors facilitates a better understanding of the user's situation and can improve the quality of HRI [2]. User pose estimation [3] based on the data obtained from the sensors can increase the ability of the robot to make appropriate decisions when interacting with the user. Thus, it can increase the robot autonomy. User privacy, however, may be compromised when robot sensors monitor the user and process their data [4]. Despite the fact that collecting detailed data from users can help to improve the quality of HRI, it can cause significant privacy concerns. Therefore, user privacy should be taken into account when developing robot sensing systems.

Using privacy-preserving sensors, e.g., thermal cameras [5] or non-vision-based sensors such as LIDAR [6] instead of commonly-used robot sensors like RGB cameras to monitor humans can help increase user privacy. The use of thermal cameras, however, as privacy-preserving way of user data is still questionable [7].

Local processing of sensors' data is another method of addressing user privacy in HRI in accordance with General Data Protection Regulation [8]. By local processing and storage, the user's data is not transferred to cloud servers far from the robot. As a result, there is a lower risk of unintended people accessing the user's data, which enhances the privacy of the user. In order to accomplish this, it is necessary to develop algorithms that can be run on a robot's typical processor. For instance, to develop DL algorithms for sensor/data fusion on the robot, lightweight models would be preferable to models that require high computational costs. Nevertheless, the downside of increasing privacy is that user monitoring may become less accurate as a result. Accordingly, this problem should be investigated from a privacy-utility trade-off perspective [9].

This paper presents a novel privacy-preserving sensing system to address the privacy-utility trade-off for real-time monitoring of the user in HRI. In this system, a thermal camera and a depth sensor, as privacy-preserving sensors, are integrated to track the user's movements when interacting with the robot. The user's 2D pose is first obtained by processing the thermal images using the Openpose library [10]. The 3D pose of the user will then be determined by fusion of the thermal camera and the depth sensor.

In the work presented in this paper, a DL model for the estimation of the user's future pose has also been developed to enhance HRI performance by improving the robot's perception of the user's situation in the future. The prediction model can also compensate for the lag time of the sensing system in monitoring the user, caused by using the thermal camera and processing its images with Openpose. The proposed model is a Seq2Seq mechanism [11] that predicts the 3D positions of the user's body joints in the following time steps by getting the current and previous joints' 3D positions as input. As part of the development of this model, the local processing problem on the robot's

[1]Adel Baselizadeh is with Department of Informatics, University of Oslo, Oslo, Norway adelb@ifi.uio.no

[2]Weria Khaksar is with Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Oslo, Norway weria.khaksar@nmbu.no

[3]Md Zia Uddin is with Department of Sustainable Communication Technologies, SINTEF Digital, Oslo, Norway and Department of Informatics, University of Oslo, Oslo, Norway zia.uddin@sintef.no

[4]Diana Saplacan is with Department of Informatics, University of Oslo, Oslo, Norway dianasa@ifi.uio.no

[5]Jim Torresen is with the Department of Informatics and RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion, University of Oslo, Oslo, Norway jimtoer@ifi.uio.no

Manuscript 325 submitted to 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). Received March 17, 2023.

processor has been taken into consideration.
Accordingly, the contributions of this paper are as follows:

1) Introducing a novel privacy-preserving sensing system for real-time user monitoring in HRI by combining thermal and depth cameras and using Openpose.
2) Developing a modified Seq2Seq model for user pose prediction by considering the local implementation of the model on the robot through reducing the processing time of the model.

Also, the performance of the proposed sensing system in determining and predicting user pose has been experimentally evaluated through practical tests, including HRI scenarios. The next parts of the paper are organized as follows:

Section II presents the details of the sensing system, including the hardware and the method used for the fusion of the data from the thermal camera and depth sensor for extracting the user's current pose. A detailed description of the prediction model is provided in section III. Section IV explains the test setup for the experimental evaluation of the sensing system and the prediction model. In section V, the results of the evaluation tests are discussed in detail. Finally, section VI concludes the paper with suggestions for future work.

II. USER MONITORING AND POSE DETECTION

This section describes the proposed sensing system for user pose detection. This sensing system protects the user's privacy while monitoring their interaction with the robot.

Determining the user's gestures and body limb positions is crucial in HRI. Proper implementation of HRI tasks and ensuring user/robot safety during the execution of these tasks require accurate tracking of the position of the user's body limbs [12]. RGB-D cameras are the most common way to determine the user's body pose in HRI or for other purposes, such as human activity monitoring. The 3D positions of the user's body joints are obtained from the images, and their body skeleton is reconstructed using this data [13]. The depth information of each joint is determined by a depth camera. Positions of joints in the other two dimensions are typically extracted by processing RGB images using libraries like Openpose [10] and AlphaPose [14].

As mentioned in the introduction, user privacy can be compromised using RGB cameras. In this work, instead of using RGB-D cameras to monitor the user's movements, a thermal and a depth camera are incorporated with the objective of enhancing user privacy in HRI. In this regard, the Openpose library is used to detect the user's body joint positions from the images taken by the thermal camera in real-time. So far, the Openpose library has only been used to process thermal images offline for human activity recognition purposes [15], not in real-time human monitoring. As a result of applying Openpose to thermal images, the user's 2D body joints positions are determined. The depth information of each joint is retrieved from the depth sensor. To accomplish this, a multi-sensor calibration, including the thermal camera and the depth sensor is performed. The calibration process consists of two steps,

1) Converting the pixel position of joints in thermal images to their corresponding pixels in depth images.
2) Calculating the joints' positions in the world coordinate based on their pixel locations in thermal images.

To perform the conversion between thermal and depth sensor images in the first step, the Homography method [16] based on the Pseudo-inverse algorithm [17] has been used.
The pixel position in the thermal image could be converted to the pixel position in the depth image using Eq. 1.

$$\begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = H^d_{th} \begin{bmatrix} u_{th} \\ v_{th} \\ 1 \end{bmatrix} \tag{1}$$

where H^d_{th} is the Homography matrix of transformation from thermal camera to depth sensor.
It should be noted that depth information is obtained from the depth sensor used in an RGB-D camera. The coordinate systems of the depth sensor and the RGB camera in the available RGB-D camera are the same.

The pseudo-inverse method was used to calculate the optimal Homography matrix H^d_{th} in Eq. 1. For this aim, the movements of an individual were recorded using RGB and thermal cameras. Openpose was then applied to the recordings of the cameras, resulting in the determination of 25 human body joints' pixel positions. In total, 483250 pixel positions of the body joints from the recordings of each camera were obtained. These body joint pixel positions were then applied to Eq. 1 using Pseudo-inverse method to calculate the Homography matrix as follows:

$$U^d_N = H^d_{th} U^{th}_N \tag{2}$$

$$U^d_N = \begin{bmatrix} u^1_d & u^2_d & ... & u^N_d \\ v^1_d & v^2_d & ... & v^N_d \\ 1 & 1 & ... & 1 \end{bmatrix}, U^{th}_N = \begin{bmatrix} u^1_{th} & u^2_{th} & ... & u^N_{th} \\ v^1_{th} & v^2_{th} & ... & v^N_{th} \\ 1 & 1 & ... & 1 \end{bmatrix} \tag{3}$$

$$U^d_N = H^d_{th} U^{th}_N \longrightarrow U^d_N (U^{th}_N)^T = H^d_{th} U^{th}_N (U^{th}_N)^T \tag{4}$$

$$H^d_{th} = (U^d_N (U^{th}_N)^T)(U^{th}_N (U^{th}_N)^T)^{-1} \tag{5}$$

where N is equal to 483250. Since the RGB and depth cameras have a similar coordinate system, the Homography matrix between the thermal camera and the depth sensor would be the same. Thus, by having the pixel position of each point in the thermal image, the corresponding pixel in the depth image is determined and the depth information of that point will be found. The second step would be the conversion of the pixel positions to the world coordinate system (X, Y, and Z) which is done using the Pinhole camera projection method [18] based on the following equations.

$$X = (u^{th} - c_x)\frac{Z}{f_x} \quad , \quad Y = (v^{th} - c_y)\frac{Z}{f_y} \quad , \quad Z = Z \tag{6}$$

where c_x and c_y are camera offsets, f_x, and f_y are camera focal parameters, and Z is the distance given by the depth image. Thus, the 3D position of each joint is calculated in real-time by integrating the thermal camera and the depth sensor. This work focuses only on the user's upper-body

Manuscript 325 submitted to 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). Received March 17, 2023.

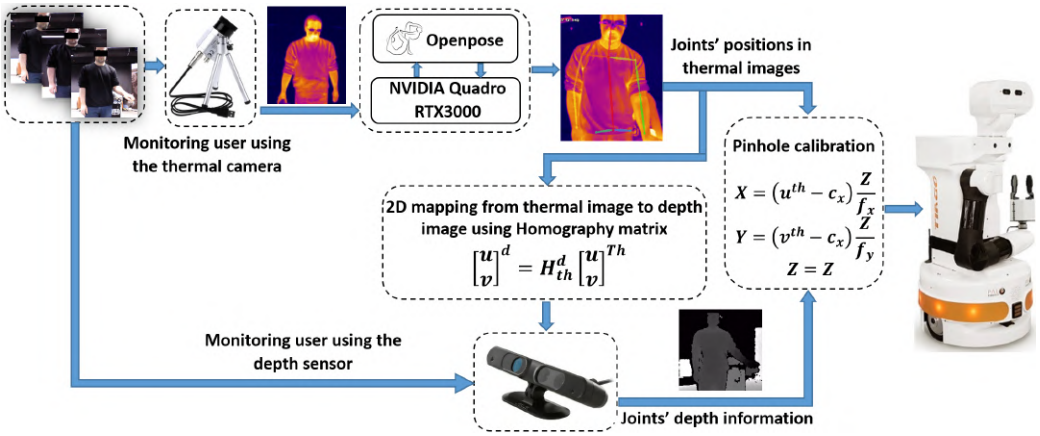

Fig. 1. The proposed sensing system for human pose detection using thermal camera and depth sensor.

limbs. Accordingly, the position of the user's eight upper-body joints, including each shoulder, elbow, and wrist joints, and the head and neck, is determined. Fig. 1 shows the proposed sensing system for human pose detection.

In the sensing system, an Optris P1450 thermal camera and Orbbec Astra S depth sensor are used. The two sensors monitor the user at a rate of up to 30 FPS. Openpose runs on a NVIDIA Quadro RTX 3000 GPU with 6 GB RAM, resulting in the extraction of the user's body joint positions at a rate of 13-15 FPS. Thus, the joint position data is determined at a maximum rate of 15 FPS, even though the sensors monitor the user at 30 FPS. It is about 20% slower than the rate at which joints' data is obtained by applying Openpose to RGB images, which is 15-18 FPS.

III. USER BODY POSE ESTIMATION

Accurate estimation of the user's poses and motions is an issue of significant importance in enhancing the machine's comprehension of human behavior. In HRI, it can contribute to a safer and more efficient execution of cooperative tasks between the user and the robot.

To enhance the quality of user monitoring and increase the robot's perception of the user's condition through the proposed sensing system, a model for predicting the user's position and movements has been developed. A further benefit of the prediction model is to compensate for the lag time caused by using a thermal camera instead of an RGB camera and applying Openpose to thermal images, as mentioned in II. The developed model is applied to the user's body joints 3D positions obtained from the sensing system. Local processing of user data has been important in developing this model.

Human pose prediction is accomplished using a variety of methods [19]. Using Kalman filters [20] and bilinear space-time basis models [21] are among the traditional techniques that are employed for this purpose. Modern human pose estimation algorithms rely largely on machine learning and deep learning models. Recurrent Neural Networks (RNN) [22] and Convolutional Neural Networks (CNN) [23] are two of the most commonly-used DL-based approaches for addressing this problem.

In this work, the Seq2Seq model has been applied for user pose prediction. Seq2Seq is an encoder-decoder-based framework that is widely used in machine translation [24]. In Seq2Seq, the encoder receives the input data and generates an internal representation of this data. This internal state is fed into the decoder, which makes predictions based on maximum likelihood estimation [25]. In this work, to use Seq2Seq to predict the user's pose, a sequence of the 3D positions of the user's body joints at the current and previous time steps is sent to the encoder as input, and the decoder predicts the joints' 3D positions in the future time steps. The user's body pose will then be reconstructed using the predicted body joint positions. The main elements of the Seq2Seq framework are the RNN units in the encoder and decoder, which are either LSTM [26] or GRU [27] cells. Using more RNN units can help Seq2Seq perform better. However, due to their recursive structure, RNN units are computationally expensive. Consequently, using too many RNN units increases the model's run time, and we need powerful processors or cloud computing solutions to execute the model. Lowering the model's computational cost enhances user privacy by reducing reliance on cloud computing.

The Seq2Seq prediction model proposed in this paper has been developed with a focus on reducing computational costs to enhance user privacy. To accomplish this, the number of RNN units (GRUs are used in the Seq2Seq model, as they generally have a lower computational cost than LSTMs [28]) in the Seq2Seq model has decreased. It should be noted, however, that reducing the number of RNNs may result in a significant reduction in the model's accuracy.

To achieve an optimal trade-off between computational cost and accuracy, the Seq2Seq architecture has been modified by incorporating established and widely-used techniques in DL. In particular, five modifications have been applied to the Seq2Seq model, as mentioned below. Each of these techniques has already been used in the Seq2Seq model alone, but in this work, they have been applied simultaneously to this structure.:

1. Residual connections It has been shown that adding a residual connection between the input and output of each RNN unit in the decoder can help improve the accuracy of

Manuscript 325 submitted to 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). Received March 17, 2023.

the prediction in the Seq2Seq model [29].

2. Attention mechanism Using an attention layer can significantly improve the accuracy of DL networks like the Seq2Seq framework [30]. The main idea behind the attention mechanism in the seq2seq model is that for each prediction, the model only uses parts of the input that includes the most relevant information instead of the entire input sequence. In this work, the Bahdanau attention mechanism [30] has been used. In the attention layer, first, the degree of correlation at time-step t of the decoder is calculated using Eq. 7.

$$e_{t,j} = a(s_{t-1}, h_i) = V^T tanh(W[h_i; s_{t-1}]) \tag{7}$$

where h_i is the i^{th} hidden state vector of the encoder, S_{t-1} is the hidden state of the decoder at time $t-1$, and V is a weight vector. In the next step, the softmax normalization operation is applied to e_{ti}, resulting in obtaining the weight α_{ti} of the output vector,

$$\alpha_{t,j} = Softmax(e_{t,i}) \tag{8}$$

The attention distribution is then calculated using Eq. 9:

$$C_t = \sum_{i=1}^{T} \alpha_{t,j} h_i \tag{9}$$

Finally, the state and the output of the decoder at time T are calculated using Eq. 10.

$$s_t = f(y_{t-1}, s_{t-1}, C_t) \quad , \quad y = g(y_{t-1}, s_t, C_t) \tag{10}$$

3. Bi-directional RNN units Bi-directional RNN (Bi-RNN) is a sequence processing model using two RNNs, one of which takes input from a forward direction and one from a backward direction. It has information from the past and the future of every point in the sequence [31].

4. Time2Vec Time2Vec is an embedding (vector representation) of time. It is a vectorized representation of time data that can be integrated with different DL-based architectures, such as the Seq2Seq mechanism, helping to improve their performance [32]. Time2Vec embedding is able to detect periodic and non-periodic patterns in time data regardless of the time scale. Assuming a scalar time τ, Time2Vec of τ is a vector of size $k+1$, denoted by $\mathbf{t2v}(\tau)$ as:

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i & \text{if } i = 0 \\ \mathrm{F}(\omega_i \tau + \phi_i) & \text{if } 1 \leq \mathrm{i} \leq k \end{cases} \tag{11}$$

where $\mathbf{t2v}(\tau)[i]$ represents the i^{th} element of t2V(t), F represents the periodic activation function (usually sin and cos functions), and ω_i and ϕ_i are learnable parameters. The linear term for $k = 0$ captures non-periodic patterns in the time data. Although Time2Vec was originally proposed as an embedding of time data, it could also be used to detect periodic patterns unrelated to time data. As an example, [33] has shown that Time2Vec can help increase pedestrian trajectory prediction accuracy using the Seq2Seq model. In order to capture periodic patterns in pedestrian trajectory, Time2Vec has been applied to input sequence data, i.e., pedestrian position data, before being sent to the encoder of the Seq2Seq model. Similarly, in this work, Time2Vec was applied to the user body joint data before being sent to the Seq2Seq mechanism.

5. Beam search The beam search algorithm can be applied to many deep learning models, such as Seq2Seq, as a final decision-making layer for selecting the best output based on maximum likelihood probability [34]. Fig. 2 shows the modified Seq2Seq architecture used for human pose prediction.

The modifications mentioned above have been applied to the original Seq2Seq framework step by step. Modified models have been compared with each other in terms of accuracy and computational cost (run time of the model in the testing phase). The model is fed a sequence of 3D positions of the user's body joints in the past two seconds as input and predicts the joints' 3D positions one second later. The INHARD dataset [35] is used to train the models. This dataset includes the 3D position of 17 human body joints collected from 16 subjects at 120 FPS when performing 13 industrial HRI tasks in cooperation with a robotic manipula-

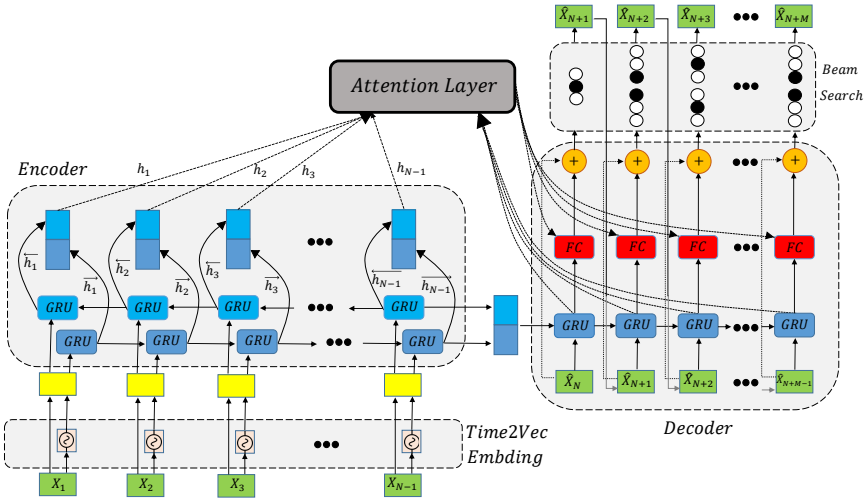

Fig. 2. The architecture of the human pose prediction model. It is a Seq2Seq model with five modifications as mentioned in the text. X_i represents the human body joint position vector at time step i. h_i is the hidden state vector i. FC shows the fully connected layer.

tor. As the developed model predicts the user's body pose in HRI, the INHARD dataset that includes human movements when interacting with a robot is a suitable alternative for training the model. All models were trained and tested on a NVIDIA Quadro RTX 3000 GPU with 6 GB RAM. As mentioned earlier, the joints' 3D position information is sent to the prediction model at a rate of about 15 FPS. To increase the accuracy of real-time prediction, all models have been trained using a subset of the INHARD dataset sampled at a rate of 15 FPS. All models have been trained using the Adams optimizer with a 0.005 learning rate. A total of 3000 iterations are performed during the training phase. The batch size for training all models is 8. For the fully connected layers in the decoder of all models, a dropout of 0.1 has been applied. Except for the original Seq2Seq model with two layers, all modified models have only one layer in the encoder and the decoder. All models have been trained using the mean squared error loss function. Table 1 shows each model's run time to achieve the prediction accuracy of 85% when testing the trained model. Compared to the other models, the model including the beam search algorithm had a much longer run time. Thus, this model has been excluded from the comparison table. Results show that the Seq2Seq model with residual connections, the attention mechanism, and the Time2Vec embedding with an embedding size of 64 has the lowest computational cost and the fastest run time in the testing phase. To achieve a specific accuracy of 85% when running the model in real-time, this model is 34% less computationally expensive than the Seq2Seq architecture without any modifications. One should note that all models have short processing times and could probably be applied in real-time human pose prediction. However, when it comes to using the model in HRI scenarios, including real-time control of the robot, shorter processing time can significantly enhance the quality of the robot control and the HRI. Therefore, the model with the fastest processing time has been applied to the practical evaluation tests.

IV. EXPERIMENTAL EVALUATION TESTS

Experimental tests have been carried out to assess the accuracy of the developed sensing system and prediction model in tracking and predicting user pose in HRI. The objectives of these tests are to determine,

1) If the proposed privacy-preserving sensing system can appropriately track and predict the user's body pose while interacting with a robot.
2) If the prediction of the user's pose using the developed prediction model can help increase the efficiency and safety of HRI.

To accomplish this, two types of evaluation tests have been conducted. The TIAGo robot is used for performing the evaluation tests as shown in Fig 1.

The first test is a trajectory-tracking scenario, in which the robotic manipulator follows the user's wrist when moving their hand at medium and fast speeds. The position of the user's wrist sent to the robot, as the desired position of the robot's end-effector to reach, is either the current or the future position of the wrist estimated by the prediction model. A comparison is made between the accuracy of human hand-following when the desired position of the robot is the current or the future position of the user's wrist. This comparison can show the effectiveness of user movement prediction on the efficiency of HRI. The test is conducted with fast and medium hand movements. To increase the test reliability, the tests are designed so as to be repetitive and the user's hand follows the same trajectory every time. The UR5 robot has been used to create repetitive motions for the hand during the evaluation tests. The user is asked to take UR5's end-effector. By creating repetitive motions for the UR5's end-effector, one can ensure that the user's hand follows the same trajectory at different test steps (Fig. 3).

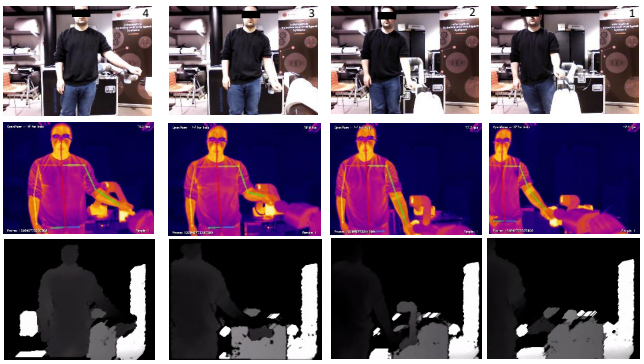

Fig. 3. The hand-following experimental setup. The user is monitored using the thermal camera and the depth sensor when taking the end-effector of the UR5 robot. By creating repetitive motions for the UR5, the trajectory of the user's hand would be the same for all tests.

The second test investigates the effectiveness of predicting

TABLE 1

COMPARISON OF RUN TIME IN THE TESTING PHASE AMONG DIFFERENT MODIFIED SEQ2SEQ ARCHITECTURES.

Model	Number of GRUs	Model parameters	Model's run time at testing phase for up to 1s prediction (ms)		
			330 ms	720 ms	1000 ms
Seq2Seq	7200	2[1]	0.006272 ± 0.00094	0.007623 ± 0.00099	0.008644 ± 0.00144
RC[2] + Seq2Seq	2600	—	0.005643 ± 0.00213	0.0069342 ± 0.00154	0.0076896 ± 0.00099
RC + Bi-RNN + Seq2Seq	1920	—	0.005461 ± 0.00074	0.006581 ± 0.00091	0.006924 ± 0.00207
AM[3] + RC + Bi-RNN + Seq2Seq	1050	—	0.005250 ± 0.00090	0.006188 ± 0.00185	0.0063041 ± 0.00701
Tim2Vec + AM + RC + Bi-RNN + Seq2Seq	800	32[4]	**0.004688 ± 0.00109**	0.005381 ± 0.00163	0.005731 ± 0.00099
	800	64[4]	0.004702 ± 0.00115	**0.005234 ± 0.00111**	**0.005624 ± 0.000950**
	1000	128[4]	0.004810 ±0.00107	0.005401 ±0.00094	0.005840 ±0.00082

[1]Number of layers [2]Residual Connection [3]Attention Mechanism [4]Time2Vec Embedding size

Manuscript 325 submitted to 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE). Received March 17, 2023.

user movements in increasing HRI safety through simple human-robot collision scenarios. This test includes two parts. In the first part, the user moves their hand towards the robot end-effector when a collision avoidance controller is activated. The controller receives the user's body pose data from the sensing system. The test consists of two steps, with the current position of the hand and its predicted position being sent to the controller in two steps. As soon as the controller detects the user's hand entering the robot's danger zone, the robot will react to avoid a collision with the hand. The robot's reaction time will be compared when it receives either the current or the predicted user's hand position.

In the second part, the user is assumed to have already entered the robot's danger zone and to be moving their hand away from the robot now. As soon as the user's hand exits the danger zone, the collision avoidance controller is deactivated, and the robot continues performing tasks. The robot's reaction time to the exit of the user's hand from the danger zone is measured. A comparison is made between the reaction times when the hand's current and predicted position is sent to the controller. Predicting the user's movements is expected to lead to faster robot reactions, resulting in safer robot performance. The user's hand movement velocity should be the same at each test step to make the robot's reaction times comparable. To move the user's hand with a constant velocity, like the hand-following test, the UR5 robot has been applied to human-robot collision tests (Fig. 4).

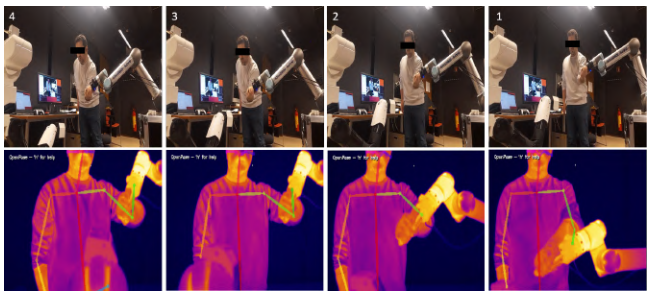

Fig. 4. Human-Robot collision test setup. The user takes the UR5's end-effector. UR5 moves the hand towards and away from the robot in the first and second tests, respectively. The output of the depth sensor is not shown in the figure.

Since the user's hand in all tests is moved using the UR5 robot, its motions would be independent of the test subject. Thus, all tests have been conducted only on a single person. However, to study the reliability and robustness of tests, methods like moving the hand at different speeds, repeating the tests several times, and moving the hand in different directions in the robot's workspace have been taken into account. The details of these methods and the results of the tests are presented in section V.

V. RESULTS

In this section, the results of the experimental tests have been reported.

- User hand-following tests

As mentioned earlier, hand-following tests are conducted at medium and fast speeds of the user's hand movement. Medium and fast motion of the hand is achieved by moving

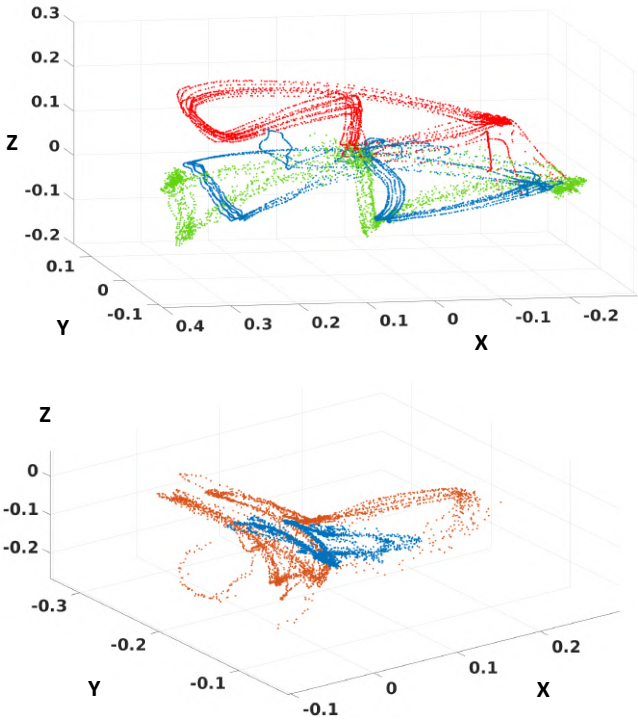

Fig. 5. Top - The trajectories of the user's hand (green) and the robot's end-effector when the prediction model is activated (blue) and deactivated (red). Below - Distribution of the trajectory tracking error when using (blue) and not using (brown) the prediction model. Dimensions are in meters.

UR5's end-effector at 30% and 60% of its maximum velocity, respectively. The trajectory of the user's hand includes motions in 3D Cartesian space. The hand-following experiment consists of four tests: moving the hand at two speeds and taking feedback from either the current or predicted hand position. The user's hand position is estimated for up to one second, and prediction results for the following 0.3 seconds are sent to the robot. In order to ensure the reliability of the results, each test was repeated ten times. Fig. 5-top shows the trajectories of the user's hand and the robot's end-effector. Fig. 5-below illustrates the distribution of tracking error when the current and predicted positions of the hand are used as reference trajectories of the robot, respectively. For brevity, only the test results at medium speed have been presented in Fig. 5. Table 2 presents the Root Mean Square Error (RMSE) of hand-following for all tests calculated using Eq. 12. The overall error and the error in each direction have been reported in Table 2. To calculate the error in each direction, the other two directions are removed from Eq. 12.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}[(X_i^h - X_i^r)^2 + (Y_i^h - Y_i^r)^2 + (Z_i^h - Z_i^r)^2]}{N}} \quad (12)$$

Where X_i^r, Y_i^r, and Z_i^r are the robot's 3D Cartesian end-effector positions, and X_i^h, Y_i^h, and Z_i^h are the hand's positions. N is the number of time steps in the test.

The hand-following test results reported in Table 2 show that the prediction model can improve the overall trajectory-following accuracy by about 25% and 10% at medium and fast speeds, respectively. A lower accuracy at fast speed is due to the prediction model being trained using human

TABLE 2
RMS OF THE TRAJECTORY TRACKING ERRORS IN THE HAND FOLLOWING TEST IN X, Y, AND Z DIRECTION AND OVERALL.

Direction	Speed	Pose prediction?	RMSE	Improvement(%)
X	Medium	Yes	0.0727	18.31
		No	0.089	
	Fast	Yes	0.0647	7.17
		No	0.0697	
Y	Medium	Yes	0.048	24.05
		No	0.0632	
	Fast	Yes	0.0625	-7.76
		No	0.058	
Z	Medium	Yes	0.0488	6.51
		No	0.0522	
	Fast	Yes	0.0521	23.83
		No	0.0684	
Overall	Medium	Yes	0.2244	24.97
		No	0.2991	
	Fast	Yes	0.3329	9.83
		No	0.3692	

usual movements, which are much closer to the user's hand movements at medium speed. According to the results of the fast speed test in the Y direction, predicting the user's hand movements negatively impacts trajectory following accuracy by 7.7%. This is probably due to the prediction model's inaccuracy in estimating the movements of the user at fast speeds. It should also be noted that during the experiments, we discovered that trajectory tracking is less efficient in the Y direction compared to the X and Z directions. It may be because the position of the user's hand in the Y direction is determined by the depth sensor, while in the X and Z directions, it is obtained from the thermal camera. The prediction of user poses also compensates for lag time caused by processing thermal images using Openpose. Compared to RGB cameras, this processing lag time is more noticeable for thermal cameras.

- Robot-user collision tests

The danger region of the robot in the human-robot collision tests is within 30 cm of the robot manipulator. Both parts of the collision avoidance tests consist of four steps, each repeated at least six times to enhance test reliability. These four parts include moving the hand in the XZ, XY, and YZ planes and in 3D space. As mentioned earlier, in the first and second parts, the hand moves toward and away from the manipulator, respectively. The robot's reaction times to avoid possible collisions with the user's hand when moving toward the manipulator are presented in Table 3.

TABLE 3
THE REACTION TIMES OF THE ROBOT TO POSSIBLE COLLISIONS AND THEIR IMPROVEMENTS DUE TO PREDICTING USER POSE.

Space	Pose prediction?	Reaction time(s)	Improvement(%)
XY	Yes	6.03	15.03
	No	5.23	
XZ	Yes	2.30	12.20
	No	2.05	
YZ	Yes	2.23	8.03
	No	2.07	
XYZ	Yes	3.70	13.84
	No	3.25	

Table 3 shows the improvements in the robot's reaction time resulting from the prediction of the user's hand movements. The results of the two parts of the human-robot collision tests indicate that predicting the user's hand position, on average, reduces reaction time by up to about 15%.

Table 4 presents the robot's reaction times to the exit of the user's hand from the danger zone and the improvement of the reaction time as a result of user pose prediction. Like hand-following tests, the predicted hand position data for 0.3 seconds later are sent to the robot in the relevant tests.

TABLE 4
THE REACTION TIMES OF THE ROBOT TO THE EXIT OF THE USER'S HAND FROM THE DANGER ZONE.

Space	Pose prediction?	Reaction time(s)	Improvement(%)
XY	Yes	4.80	11.10
	No	5.40	
XZ	Yes	3.57	14.06
	No	4.15	
YZ	Yes	2.11	17.58
	No	2.56	
XYZ	Yes	2.95	8.76
	No	3.23	

The robot's reaction time improvement is seen in the second part of the human-robot collision test when the user moved their hand away from the robot. Based on the results reported in Table 4, the robot can react up to 17% faster when it detects the user's hand is getting out of the danger zone. Thus, the robot can start performing the tasks 17% earlier by providing it with the user's predicted hand position data.

VI. CONCLUSIONS

In this paper, a robot-based privacy-preserving sensing system for the user's upper-body motion monitoring in HRI using the integration of a thermal camera and a depth sensor and using Openpose has been developed. A lightweight DL model has also been trained to predict user pose when interacting with the robot and being monitored using the proposed sensing system.

The prediction model is based on a Seq2Seq architecture. To make the model computationally optimized, its architecture has been modified by adding four DL-based units, i.e., attention layer, residual connections, Bi-RNNs, and Time2Vec embedding. It is shown that by applying these add-ons simultaneously to the Seq2Seq mechanism, the computational cost of the model is significantly reduced while its accuracy remains unchanged.

The proposed sensing system and the prediction model have been evaluated through experimental tests. In these tests, the effectiveness of predicting user poses has been studied in a HRI setting when they are monitored using the available sensing system. The results indicate that HRI can be made more efficient and safe by using the available privacy-preserving monitoring platform and prediction model. Predicting the user's body pose provided the robot with information about the future posture of the user, leading to improvement of the HRI performance in terms of safety and efficiency. Moreover, it compensated for the lag time in

the sensing system caused by the processing of the thermal camera image data using Openpose.

The accuracy of the prediction model can be improved by training it on other public or specifically collected datasets. Using alternative DL libraries for human joint detection in thermal images, e.g., AlphaPose [14], may also improve the performance of the sensing system. Furthermore, the Seq2Seq mechanism can be improved by incorporating other deep learning techniques, such as greedy search [36].

In future work, the proposed sensing system and prediction model will be evaluated in complex human-robot interaction scenarios, including tasks that require collaboration between humans and robots, with advanced collision avoidance controllers present. Also, another topic that could be taken into account is using other privacy-preserving sensors, like non-vision-based sensors in the sensing system, whcich can help to increase the robot's perception of the user's condition.

ACKNOWLEDGMENT

This work is partially supported by The Research Council of Norway as a part of the Vulnerability in the Robot Society (VIROS) project, under grant agreement 288285, Predictive and Intuitive Robot Companion (PIRC) project under grant agreement 312333 and through its Centers of Excellence scheme, RITMO with the project no. 262762.

REFERENCES

[1] Jenay M Beer, Arthur D Fisk, and Wendy A Rogers. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction*, 3(2):74, 2014.

[2] Francesco Semeraro, Alexander Griffiths, and Angelo Cangelosi. Human–robot collaboration and machine learning: A systematic review of recent research. *Robotics and Computer-Integrated Manufacturing*, 79:102432, 2023.

[3] Yalin Cheng, Pengfei Yi, Rui Liu, Jing Dong, and Dongsheng Zhou. Human-robot interaction method combining human pose estimation and motion intention recognition. In *24th International Conference on Computer Supported Cooperative Work in Design*, 2021.

[4] Matthew Rueben and William D Smart. Privacy in human-robot interaction: Survey and future work. *We robot*, 2016:5th, 2016.

[5] Laura Boccanfuso, Quan Wang, Iolanda Leite, Beibin Li, Colette Torres, Lisa Chen, Nicole Salomons, Claire Foster, Barney, et al. A thermal emotion classifier for improved human-robot interaction. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 718–723. IEEE, 2016.

[6] Dražen Brščić, Rhys Wyn Evans, Matthias Rehm, and Takayuki Kanda. Using a rotating 3d lidar on a mobile robot for estimation of person's body angle and gender. *Sensors*, 20(14), 2020.

[7] Naomi Lintvedt. Thermal imaging in robotics as a privacy enhancing or privacy invasive measure? the necessity of a holistic approach to privacy in human-robot interaction. *Zagreb, Croatia*, page 12, 2022.

[8] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[9] Christoph Lutz, Maren Schöttler, and Christian Pieter Hoffmann. The privacy implications of social robots: Scoping review and expert interviews. *Mobile Media & Communication*, 7(3):412–434, 2019.

[10] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[11] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[12] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 5(4):261–349, 2017.

[13] Mercedes Garcia-Salguero, Javier Gonzalez-Jimenez, and Francisco-Angel Moreno. Human 3d pose estimation with a tilting camera for social mobile robot interaction. *Sensors*, 19(22), 2019.

[14] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[15] Md. Zia Uddin, Weria Khaksar, and Jim Torresen. A thermal camera-based activity recognition using discriminant skeleton features and rnn. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 777–782, 2019.

[16] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, 2007.

[17] Richard Bouldin. The pseudo-inverse of a product. *SIAM Journal on Applied Mathematics*, 24(4):489–495, 1973.

[18] Kenneth M Dawson and David V. Simple pinhole camera calibration. *International Journal of Imaging Systems and Technology*, 1994.

[19] Kedi Lyu, Haipeng Chen, Zhenguang Liu, Beiqi Zhang, and Ruili Wang. 3d human motion prediction: A survey. *Neurocomputing*, 489:345–365, 2022.

[20] Tommaso Lisini Baldi, Francesco Farina, Andrea Garulli, Antonio Giannitrapani, and Domenico Prattichizzo. Upper body pose estimation using wearable inertial sensors and multiplicative kalman filter. *IEEE Sensors Journal*, 20(1):492–500, 2019.

[21] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics (TOG)*, 31(2):1–12, 2012.

[22] Jianjing Zhang, Hongyi Liu, Qing Chang, Lihui Wang, and Robert X Gao. Recurrent neural network for motion trajectory prediction in human-robot collaborative assembly. *CIRP annals*, 69(1):9–12, 2020.

[23] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, pages 332–347. Springer, 2015.

[24] Gaurav Tiwari, Arushi Sharma, Aman Sahotra, and Rajiv Kapoor. English-hindi neural machine translation-lstm seq2seq and convs2s. In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pages 871–875. IEEE, 2020.

[25] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[26] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[27] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems*, pages 1597–1600. IEEE, 2017.

[28] Shudong Yang, Xueying Yu, and Ying Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International Workshop on Electronic Communication and Artificial Intelligence*, pages 98–101, 2020.

[29] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017.

[30] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[31] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 1997.

[32] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[33] Victor Peñaloza. Time2vec embedding on a seq2seq bi-directional lstm network for pedestrian trajectory prediction. *Res. Comput. Sci.*, 149:249–260, 2020.

[34] Sam Wiseman and Alexander M Rush. Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv:1606.02960*, 2016.

[35] Mejdi Dallel, Vincent Havard, David Baudry, and Xavier Savatier. Inhard-industrial human action recognition dataset in the context of industrial collaborative robotics. In *2020 IEEE International Conference on Human-Machine Systems*, pages 1–6. IEEE, 2020.

[36] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov), 2002.