

Article

# Advanced Neural Classifier-Based Effective Human Assistance Robots Using Comparable Interactive Input Assessment Technique

Mohammed Albekairi <sup>1</sup>, Khaled Kaaniche <sup>1,\*</sup>, Ghulam Abbas <sup>2</sup>, Paolo Mercorelli <sup>3,\*</sup>, Meshari D. Alanazi <sup>1</sup>  
and Ahmad Almadhor <sup>4</sup>

<sup>1</sup> Department of Electrical Engineering, College of Engineering, Jouf University, Sakakah 72388, Saudi Arabia; msalbekairi@ju.edu.sa (M.A.); mdalsayer@ju.edu.sa (M.D.A.)

<sup>2</sup> School of Electrical Engineering, Southeast University, Nanjing 210096, China; lashariabbas@seu.edu.cn

<sup>3</sup> Institute for Production Technology and Systems (IPTS), Leuphana Universität Lüneburg, 21335 Lüneburg, Germany

<sup>4</sup> Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Sakakah 72388, Saudi Arabia; aaalmadhor@ju.edu.sa

\* Correspondence: kkaaniche@ju.edu.sa (K.K.); paolo.mercorelli@leuphana.de (P.M.)

**Abstract:** The role of robotic systems in human assistance is inevitable with the bots that assist with interactive and voice commands. For cooperative and precise assistance, the understandability of these bots needs better input analysis. This article introduces a Comparable Input Assessment Technique (CIAT) to improve the bot system's understandability. This research introduces a novel approach for HRI that uses optimized algorithms for input detection, analysis, and response generation in conjunction with advanced neural classifiers. This approach employs deep learning models to enhance the accuracy of input identification and processing efficiency, in contrast to previous approaches that often depended on conventional detection techniques and basic analytical methods. Regardless of the input type, this technique defines cooperative control for assistance from previous histories. The inputs are cooperatively validated for the instruction responses for human assistance through defined classifications. For this purpose, a neural classifier is used; the maximum possibilities for assistance using self-detected instructions are recommended for the user. The neural classifier is divided into two categories according to its maximum comparable limits: precise instruction and least assessment inputs. For this purpose, the robot system is trained using previous histories and new assistance activities. The learning process performs comparable validations between detected and unrecognizable inputs with a classification that reduces understandability errors. Therefore, the proposed technique was found to reduce response time by 6.81%, improve input detection by 8.73%, and provide assistance by 12.23% under varying inputs.

**Keywords:** neural networks; machine learning; robot systems; interactive classification; optimal control

**MSC:** 68T40; 68T07



**Citation:** Albekairi, M.; Kaaniche, K.; Abbas, G.; Mercorelli, P.; Alanazi, M.D.; Almadhor, A. Advanced Neural Classifier-Based Effective Human Assistance Robots Using Comparable Interactive Input Assessment Technique. *Mathematics* **2024**, *12*, 2500. <https://doi.org/10.3390/math12162500>

Academic Editor: António Lopes

Received: 18 July 2024

Revised: 10 August 2024

Accepted: 12 August 2024

Published: 13 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Human-robot interaction (HRI) is a technique used to understand, design, and evaluate robotic systems for humans. HRI is mainly used to enhance the performance range of boring work using robots [1]. Various input assessments are used for human interactive robots. A spatial reasoning-based situation assessment method is used in the HRI object manipulation process [2]. Spatial reasoning is used here to analyze the efficiency and optimal interaction services among humans and robots [3]. The analyzed data produce feasible data used as inputs to perform user tasks. The spatial features are detected from the datasets, minimizing the computational cost of the input assessment process [4]. The

spatial reasoning-based method increases the decision-making accuracy ratio, improving interactive services among users [5]. A quality assessment technique is used in HRI for the performance improvement process. The quality assessment technique identifies the key values and characteristics presented in the interactive services [6]. The identified values are used for further performance improvement, which minimizes the latency of the task allocation process. The assessment technique enhances the feasibility and robustness range of the HRI systems [7].

Human assistance robots use voice commands to perform tasks for their users. A voice command is generated remotely using a voice module. Human assistance robot input classification is a crucial task that identifies the exact owner of the robots [8]. It is an authentication process that reduces human assistance robots' complexity. Various classification methods are used to analyze the variables of the commands. A gesture-based human assistance classification method is used in HRI for manufacturing [9]. The classification method analyzes the variables provided via wearable and wireless sensors. The capture variables provide optimal features and segments necessary for human assistance in HRI [10]. The assistance classification method improves efficiency in performing particular user tasks. A hybrid assistance classification using a neural network (NN) is used in HRI [11]. The NN algorithm uses a classifier that classifies the users' voice and motion commands. The classified data produces feasible information that is important for interaction services. The NN-based technique improves the overall performance range of the systems [12].

Neural network (NN)-based methods are used for the input process in HRI. The NN algorithm mainly analyzes the inputs produced via human and robot interactions [13]. A NN-based approach is used for input processing in HRI. The aim is to analyze the variables containing virtual inertia inputs to process a task. The NN-based approach is designed to improve the systems' HRI cooperation and performance range [14]. The NN-based approach reduces the error and failure range in performing tasks for the users in HRI [15]. A neural integrator model is used to assist robots in input processing. The neural integrator is mainly used as a gradual accumulator that evaluates the input commands of the users to the robots [16]. The input command voices are classified based on the necessity and functionality of the process. The inputs are processed to perform particular tasks, enhancing the systems' feasibility range. The integrator model reduces the robots' cognitive workload and complexity during interactive services [17]. Human-robot interaction is a means of sharing identifiable objects from the environment via communication between humans and robots. Typically, the images obtained from the surroundings are three-dimensional, and the robot has trouble detecting them. To precisely identify the objects and produce results for humans, 3D to 2D conversion is used [18].

The main objective and contributions of the paper are:

- Designing the Comparable Input Assessment Technique (CIAT) to improve the bot's system understandability.
- Assessing the human assistance robots based on heterogeneous comparable input assessment.
- Employing the neural classifier for classifying inputs, precision instructions, and least assessments.
- The experimental outcomes have been performed, and the suggested CIAT model increases input detection and assistance and reduces the response time compared to other existing models.

The rest of the paper is followed by Section 2, which discusses related works. Section 3 proposes the Comparable Input Assessment Technique (CIAT) framework. The results and discussion are explained in Section 4, followed by Section 5, which concludes the novel idea of the study.

## 2. Related Works

### 2.1. Interactive Frameworks

Liao et al. [19] proposed an ergo-interactive framework for human-robot collaboration (HRC) and dynamic movement primitives (DMP). The actual aim of the framework is to analyze human-robot interactions. The analyzed data produces optimal features to detect the task requirement range. A mobile manipulator is used here to evaluate the ergonomic forces for the tasks. The proposed framework increases the adaptation process's accuracy, enhancing the system's feasibility level.

Hindemith et al. [20] designed an interactive robot task-learning method for HRC. The designed method is mainly used as an adaptive-evolution strategy, providing proper human teaching proficiency to the robots. An optimization algorithm is implemented to investigate users' feedback during the learning process. The designed method increases the overall performance range in the human-robot interaction process.

Qian et al. [21] introduced a new phase estimation algorithm-based model for proactive assistance in HRC tasks. The main aim of the model is to provide uniform interval interpolation and variables to perform tasks in HRC. Interactive movement primitives are provided to the robots, reducing the tasks' computational complexity. The introduced model improves the effectiveness range of HRC systems.

Di Marino et al. [22] proposed an interactive graph-based tool for HRC workplaces. The actual goal of the model is to provide proper design services to improve the strategic capacity of the robots. The proposed model is a multi-level design that provides feasible decision-making services to the robots. The proposed model enhances the performance and robustness of HRC applications.

### 2.2. Mathematical Model-Based Models

Burks et al. [23] developed an online partially observable Markov decision process (POMDP) framework for human-assisted robotic planning and sensing (HARPS). The developed framework identifies the actual structure and semantic features of HARPS. Semantic data produces necessary features that minimize the latency of resource allocation and planning processes. The developed POMDP framework improves the robustness and interaction services among human robots.

Muramatsu et al. [24] introduced a mathematical bias model-based involuntary stabilization method for human-robot interaction (HRI). The introduced method is mostly used to stabilize the involuntary behaviors of robots in applications. The mathematical model analyzes the interaction process's exact stability range, reducing the computational cost of tasks. The introduced method improves the significance ratio of the interaction services.

Fu et al. [25] proposed an android robot-based social connectedness model in HRI. The actual goal of the model is to improve the intra-group connectedness range in HRI. It also shares the recent experience of the robots in the group member conversation process. It is mainly used to address the issues that cause conversation latency. The proposed model increases the performance range for providing optimal conversation services with people.

Rückert et al. [26] designed a new biofeedback investigation method for HRI in collaborative assembly. The designed method is commonly used for stress detection from biofeedback. The designed method analyzes the endogenous signals presented in the feedback, eliminating unwanted energy consumption in the detection process. Experimental results show that the designed method reduces the overall behavioral adaptation of the robots.

Li et al. [27] developed an integrated approach for robotic sit-to-stand (STS) assistance. The developed approach uses a long short-term memory (LSTM) algorithm to identify the human's intention over the tasks. A motion capture system is used here to capture the STS assistance level of the robots. The LSTM algorithm minimizes the workload ratio, enhancing the system's feasibility level. The developed approach increases the accuracy of assisting robots in performing user tasks.

Yu et al. [28] proposed a human-robot interaction system (HRIS) with human perception and action recognition. A monocular, multi-person, three-dimensional (3D) pose estimation method is used here to estimate the interaction scenarios in the systems. The estimation method detects the exact consecutive frames of the person, increasing the stability ratio of the action recognition process. The proposed HRIS enhances the effectiveness and feasibility of the systems.

### 2.3. Deep Learning-Based Models

Zhou et al. [29] developed an attention-based deep learning (DL) approach for HRC's inertial motion recognition and estimation. The robots' exact inertial motion is recognized, producing data for the performance improvement process. The recognized data allows the robots to perform partial human tasks in HRC. The developed approach increases the accuracy of motion recognition and estimation processes.

Zhang et al. [30] designed an innovative multi-dimensional learning algorithm for HRC. A dynamic movement primitive model is used here to analyze the robots' force characteristics and position trajectory. The primitive model identifies the skill generalization and dynamic parameters used to improve the robustness level of the systems. The designed algorithm improves the robot's trajectory compliance, reducing task latency.

Lippi et al. [31] proposed a distributed framework for HRI. The proposed framework is mostly used to interact with an object-manipulated robot in HRI systems. The proposed framework uses linear quadratic tracking (LQT) with a recursive least squares (RLS) technique to estimate the robot's intention over tasks. The RLS technique analyzes the feasible data necessary for the requirement allocation process. The proposed framework improves the overall functional capabilities of the robots.

Liau et al. [32] introduced a genetic algorithm-based task allocation method using two robots for HRC. The important characteristics and capabilities of the robots are analyzed for the task allocation process. The analyzed features produce optimal information that is necessary for further processing. Compared with other methods, the introduced method increases the accuracy level of the task allocation process.

Sidaoui et al. [33] designed a joint initiative supervised autonomy (JISA) framework for HRI. The designed framework mainly evaluates the robots' self-confidence (SC) range. The JISA framework analyzes the actual demands that the users provide. The JISA framework minimizes the computation process's time and energy consumption ratio. The designed framework improves the overall accuracy range in the reconstruction process.

Ince et al. [34] introduced an audiovisual interface-based drumming system for multi-modal HRI. The introduced system is mainly used in conjunction with the robot in HRI. The users' feedback provides various information necessary for the drumming system. It is used as an evaluation tool to evaluate the gaming capacity of the robots. The introduced system enhances the robot's visual representations, increasing users' satisfaction.

Jarrah et al. [35] proposed improving process parameters for the growth of carbon nanotubes (CNTs) by the hybrid technique of response surface methodology (RSM) and bat algorithm (BA). The process successfully adjusts variables such as catalyst weight, reaction temperature, time, and methane partial pressure to maximize the yield of CNTs for mass production. Findings show that BA performs better than RSM, improving CNT yield by 21% in a single dataset while exhibiting quick and steady convergence. Additional validation is necessary in actual laboratories.

Online learning methods have become popular due to the COVID-19 pandemic's considerable impact on several sectors, including education. Shorman et al. [36] proposed Arabic language learning websites based on various criteria to help students and provide teachers with resources. Requirements, including usability, usefulness, and the caliber of the learning content, which emphasizes language proficiency in writing, reading, speaking, and listening, are evaluated numerically and qualitatively. Nevertheless, this study is not intended to cover computer systems for learning outside of websites.

Suprakas Saren et al. [37] suggested comparing alternative modalities in the context of multimodal human-robot interaction. The author used the NASA Task Load Index (TLX), a subjective, multi-dimensional scale that measures the perceived cognitive burden of a user, to assess task completion times and estimate cognitive effort. Research showed that the order of alternative modalities' completion durations was the same in both single-task and dual-task situations. Rankings according to perceived cognitive burden, however, varied. While the eye-gaze-based modality was linked to the greatest TLX score in the dual-task research, the gesture-based modality yielded the best TLX score in the single-task trial. Similarly, in the one-task trial, the speech-based modality's TLX score was lower than eye gazing and gesture, but in the two-task trial, it was in the middle of the pack. Results like this show that the effectiveness of different modalities depends on the users' choices and the scenario at hand.

Tian Wang et al. [38] proposed multimodal human-robot interaction for human-centric smart manufacturing. This paper aims to rectify this shortcoming by extensively investigating multimodal HRI, primarily focusing on the following four modalities: visual, auditory, linguistic, haptic, and physiological sensing. This discussion includes a comprehensive overview that covers algorithms, interface devices, and practical considerations. This paper delves into the three aspects of perception, cognition, and action, combining multimodal HRI with cognitive science in a new way. The goal is to demystify the algorithms that are inherent to multimodal HRI. Finally, it outlines prospective paths for multimodal HRI in smart manufacturing, emphasizing humans and highlighting the practical obstacles.

Several major drawbacks were common in earlier human-robot interaction studies. These included inaccurate detection of various user inputs, delayed and inefficient processing of inputs, and insufficient response creation, which occasionally resulted in irrelevant or incorrect assistance. The main causes of these deficiencies might be traced back to antiquated detection technology, straightforward analysis procedures, and slower processing rates. Furthermore, dependability was compromised since many early systems lacked adequate error detection capabilities. This article introduces a Comparable Input Assessment Technique (CIAT) to improve the bot system's understandability.

### 3. Comparable Input Assessment Technique (CIAT)

Human-assisted robots play a crucial role in artificial intelligence, providing cooperative control, robot control, and other task optimization. In this optimization category, the bots assist in the particular task by accompanying and observing the processing step. Here, the observation of certain task completions is based on recognizing human activities. Processing this training is aided by this approach, where it deploys to forward the necessary tasks to accomplish. This detection process estimates the coordination between the detection and unrecognition input, and reliable optimization is provided under AI since the bot performs an automated, repetitive, and pre-defined task, which is examined based on human activities. Figure 1 introduces the proposed CIA technique.

The human activities are based on human user behavior, derive the planning task, and reduce the completion time. The recognition of human activities is based on a multi-class classification. The proposed work introduces CIAT integrated with the deep neural network that performs the classification. The classification is based on a better analysis of the optimization. Thus, the scope of this work is to reduce time and error, while the precision and response factors are improved based on this deep neural network. Hence, it estimates the classification from the neural network where the new activities are recorded for the recognition phase. The ability of neural classifiers to identify and understand intricate patterns in various human inputs, including voice, gestures, and motions, is crucial for efficient human-robot interaction. Their capacity to learn and grow with more data allows the robot to react precisely to various encounters. As an advantage, neural classifiers can reliably handle inconsistent and noisy human inputs, making them ideal for real-world circumstances where human communication is not always error-free. Thus,

the preliminary step is to recognize the human-assisting robots, which include gestures, speech, and emotions, which are equated in the equation below.

$$R = \left\{ \left[ (G' + S' + E') * \frac{i_p}{b_t} \right] * \sum_{i_p}^{b_t} (G' + S') + A * \left( \frac{\prod_{E'} A}{b_t + i_p} \right) + \left( \frac{\sum \left( \frac{1}{\frac{i_p}{b_t}} \right)}{\left( \frac{A}{(G'+S'+E')} \right)} \right) * \left( \frac{\left( \frac{\sum_A b_t + i_p}{1} \right)}{\left( \frac{G'+S'}{E'} \right)} \right) \right\} \quad (1)$$

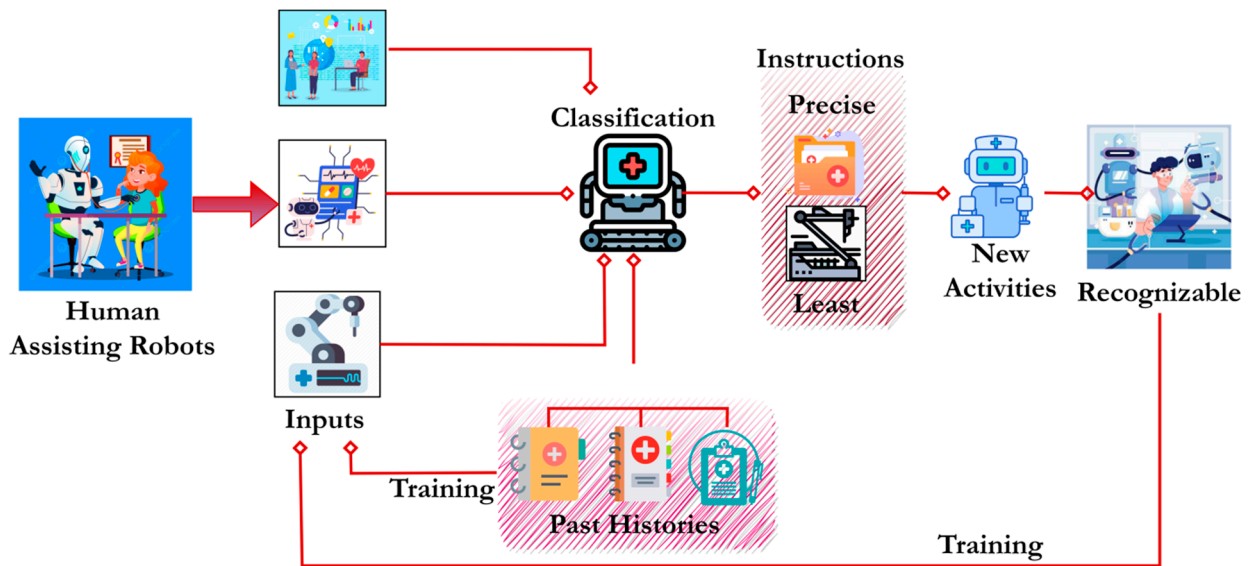


Figure 1. Proposed CIAT Technique.

The recognition is performed in this case, where the input from the human assisting robots is described as  $R$ . This stage includes the gestures, speech, and emotions of the human, and they are symbolized as  $G'$ ,  $S'$  and  $E'$ . The input is labeled as  $i_p$ , bots are  $b_t$  the acquisition of the input is  $A$ . In this case, the recognition is processed, and the appropriate output is measured from the input. Hence, the input is performed for the varying gestures it observes and stores in the database. The acquisition is performed from the database and provides reliable output. This approach indicates the bots system where the analysis is carried out for the different processing sets from the recognition.

These input-based bots are performed by acquiring gesture, speech, and emotion, and

they are described as  $\left( \frac{\sum \left( \frac{1}{\frac{i_p}{b_t}} \right)}{\left( \frac{A}{(G'+S'+E')} \right)} \right)$ . The examination is measured for the varying inputs

acquired from the robots, and based on this, the processing is carried out appropriately. By providing this, it detects the activities with the previously stored dataset and forwards the result. This mapping process indicates the history and the new activity of the bots in the AI. The optimization techniques rely on the bot's pre-defined and automatic behavior. In this case, the robotic control is used to accomplish the particular task planning and completion time at the mentioned interval, formulated below.

$$b_t(o') = (A + i_p) * \left( \frac{\sum_R (G' + S' + E')}{P_h / (n_l + m_i)} \right) + \prod_{i_p} [A + (G' + S' + E')] * \left( 1 + \frac{R + A}{\sum_{P_h} (n_l + m_i)} \right) + i_p - P_h \quad (2)$$

The robot control is derived from the above equation, where acquiring the particular input is used to recognize the bots based on human activities. The control is described as  $o'$ ,

and the history is labeled as  $P_h$ , the planning task and completion time are represented as  $n_l$  and  $m_i$ . This stage of processing is used to provide a better detection of gesture, speech, and emotion based on this control. This observation estimates the precise processing by comparing the bots and the human-assisting robots. The mapping phase is involved in this equation, where the current and previous stages are used to develop the recognition of the bots using AI. The bot control based on different inputs is represented in Figure 2.

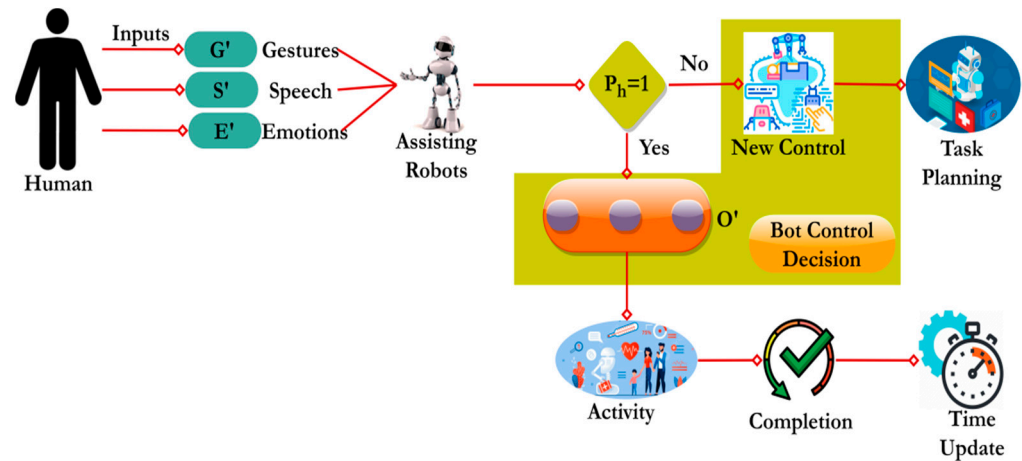


Figure 2. Bot Control Decision Based on Different Inputs.

In Figure 2, the bot control decision relies on the  $p_h$  for which previous/new activity recognition is performed. If the previous history shows up different  $o'$  then  $A = \sum(G', S', E')$  is recognized. The absent entries are defined to detect new controls. If the  $i_p$  are synchronized, then  $n_l$  is defined for individual A such that  $m_i$  is consolidated. The proposed classification requires precision, and the least  $i_p$  is detained from activity recognition. Bot control is a tool for robotics that allows robots to navigate their surroundings without human intervention. One must plan a robot route, avoid obstacles, and make decisions to travel effectively and securely. Programming and managing the execution of certain tasks, such as assembly, welding, or material handling, is what bot control is all about for industrial robots. This guarantees that robots can consistently and accurately carry out their tasks.

This control system is used to process the acquisition of input from the history and forward better results based on completion time. Hence, the completion time is used to examine the planning task and the fixed time to determine whether the task is completed or not, which involves a timely manner. The timely manner is detected in this step, where

it includes the recognition of the bots, and it is represented as  $\left( \frac{\sum_R(G'+S'+E')}{P_h / (n_l + m_i)} \right)$ . From this

computation, the analysis is forwarded to find the recognition of the bot's behavior, which maps with human activities, and it is  $[A + (G' + S' + E')]$ . By processing this, the task optimization is computed in the equation below.

$$P_z = \frac{1}{b_t(i_p)} + \max_A(o' + b_t) + D * \sqrt{\left( \frac{(A + i_p)}{\sum_R(n_l + m_i)} \right)} + \prod_A^{b_t}(i_p + D) + (G' + S' + E') * (P_h + R) \tag{3}$$

The task optimization is executed, where the mapping is performed for the robotic control used to deliver the input human assistance. The task optimization is labeled as  $P_z$ , here, it indicates the planning task and the completion time that require the maximum control for the bots that acquire the detection, and it is  $D$ . The periodic comment is calculated for the bot's control for task optimization. This technique invokes the bots to operate on perfect standards and maintain the benchmark. Examining these computation

steps for input detection is used to provide reliable recognition among bots. In Algorithm 1, the task optimization process is detailed.

---

**Algorithm 1** Task Optimization

---

Function  $P_z(i_p, o')$   
 Step 1 : Initialize R as  $\{\sum G', S', E'\}$   
 Step 2 : While  $i_p \neq 0$   
 Step 3 :  $i_p = \sum(G', S', E')$  such that  $A = i_p$   
 Step 4 : Perform  $n_l$  using  $A \forall i_p$   
 Step 5 : If  $n_l == A$   
 Step 6 : Map  $(A + \sum i_p)$  to  $P_z$   
 Step 7 :  $P_z = \max(o' + b_t)$   
 Step 8 : If  $P_z \neq \max(o' + b_t)$   
 Step 9 :  $n_l = P_h + R$   
 Step 10 : Go to Step 3  
 Step 11 : End if  
 Step 12 :  $P_z = (o' + b_t) - A$   
 Step 13 : End if  
 Step 14 : Map R to  $P_z \forall n_l < (A + i_p)$   
 Step 15 : End while

---

Based on this recognition, an error in the bot’s operation is detected. The occurrences are detected in this approach and kept before the operation, and the controls are observed for task optimization. Task optimization plays a key role in AI for the bot’s operations. This method involves the detection from the input, which is based on the recognition, and it is formulated as  $(i_p + D) + (G' + S' + E')$ . Thus, acquiring the input from the required bots is used for the forthcoming classification; for this procedure, the training optimization is proposed and equated to the equation below.

$$T_z = \prod_{P_z}^{o'} (b_t + A) * \left[ \left( \frac{(n_l + m_i)}{(D + b_t)} \right) + \left( \frac{R/P_h}{m_p - b_t} \right) \right] * \sum_A (b_t + R) * m_p - P_h + (R * b_t(o')) \tag{4}$$

Training optimization is the key concept in this paper, which establishes the link between the bots and the humans, and it is represented as  $T_z$ , mapping is described as  $m_p$ . Training optimization is used to train the error factor and reduce further computation steps. Based on this detection, the new activity is re-initialized to recognize the bots in AI appropriately. In AI, error detection is based on training optimization. In this category, the error is detected at this stage by mapping with the previous processing. From this previous processing, the mapping allows the bots to perform well in AI.

This approach delivers a better outcome for the analysis that delivers the better training phase. The optimization technique involves the bot’s improvement in planning and completing a particular task on time. From this approach, task planning indicates new activity recognition, which estimates the history of the re-initialization of the particular task. This recognition for the bots in this stage maps the current and past historical activity and delivers the result. Thus, the training optimization is carried out on this platform; after this, the input is acquired from the robot control, and task optimization is derived from the following equation:

$$A = (i_p + b_t) * \left( \frac{(T_r + o')}{\sum_R (n_l + m_i)} \right) * \prod_R (o' + b_t) * P_h \tag{5}$$

The input is acquired from the previous operation state, including task optimization and robot control. From this derivation, the input is fetched and processed to find the human assisting the activity. This activity relies on error detection, in which recognition is performed based on history. The history mapping is measured for the bot’s recognition, and the planning task indicates the training optimization. These training optimization tech-

niques measure the classification of reliable and non-reliable processes. This observation is based on the history and provides task optimization and control of the bots. The mapping of  $A$  is diagrammatically portrayed in Figure 3.

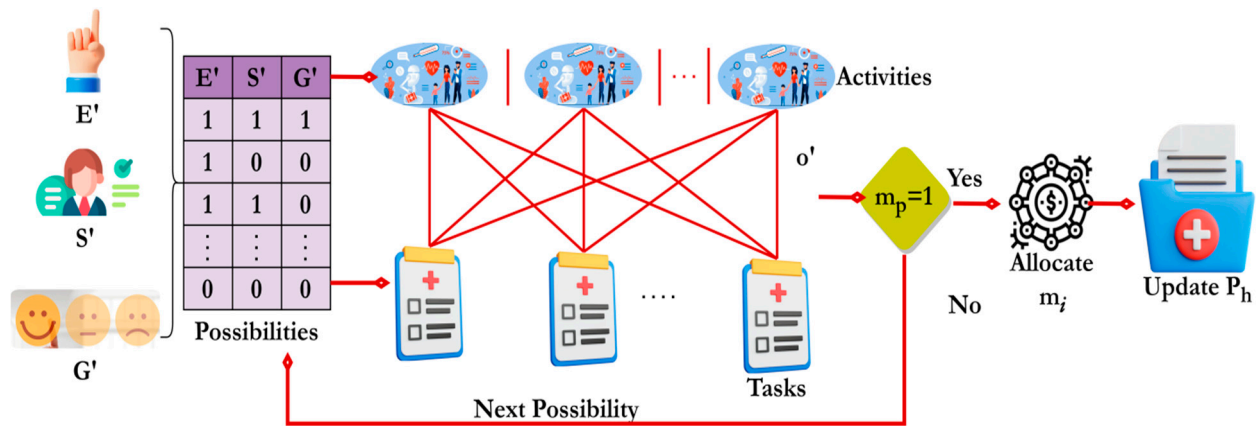


Figure 3. Mapping of  $A$ .

The mapping process is performed between input  $A$  and activities using  $o'$ . The possibilities for various tasks and their corresponding  $P_z$  define the  $n_i$  filtration. If  $m_p = 0$  then  $T_z$  is induced for update new  $o'$  from the bot. This is feasible for at least one input of the  $A$  acquired, resulting in a high analysis rate. Considering the changes in  $m_p$ , the possibilities are revised to achieve a high  $m_i$  and thus  $P_h$  are updated. The mapping is referenced for further understandability and input analysis to reduce errors (Figure 3). The input is acquired from the previous stage, indicating dependable training for the fetched input. From the input stage, the planning and completion of the task indicate the AI optimization for the bot's behavior in this paper. This establishes the optimization algorithm for the neural network that provides robot control of bots based on previous history. The history of detection is based on the recognition of the activity, which forwards the input to the next computation stage. In this approach, a human-assisting robot is given as the input, and this classification is followed up using a deep neural network.

### 3.1. Classification Using Deep Neural Networks

The classification model indicates the neural network for the precise and least positive failure. From this detection, the precise output is derived for the bots, which provides the bots with the planning task. Since it is a pre-defined model, the bots are the programs that work automatically. This working stage involves task completion and response at the mentioned time. These factors are introduced in the deep neural network and provide better understandability by proposing a classification formulated in the equation below.

$$\alpha = \begin{cases} \prod_{b_t} (R + A) * \left( \frac{P_h + b_t}{m_p / o'} \right) + D - P_z, \in \text{Precise} \\ \left( \frac{o' + b_t}{(R + A)} \right) * \sum_D (i_p + b_t) + m_i, \in \text{Least} \end{cases} \quad (6)$$

The classification is provided in this neural network, where it deploys the precision and the least positive failure and is described as  $\alpha$ . These two factors indicate the history of the bots and provide reliable classification precision with the least failure. This observation is processed in the neural classification, which holds the  $n$ -number of neurons and performs efficient detection among the input bots. Based on this, task optimization is calculated. The first case indicates the precision where the history is used to map with the current

bot’s detection, and it is formulated as  $\left(\frac{P_h + b_t}{m_p / o'}\right)$ . From this stage, the least positive failure provides the recognition acquired for the bot’s control. Thus, the classification is performed from the input acquired, and from this, the new activity recording is calculated in the equation below.

$$E(n) = \left(\frac{R + D}{\prod_{o'}(b_t * n_i)}\right) * [(m_p + e_r) + (\alpha)] * n_w + (P_h + b_t(o')) \tag{7}$$

The new activity is detected at this stage, providing reliable computation for error detection. In this scenario, the mapping is performed to detect the error factor in this step, and it is represented as  $e_r$ , the new activity is  $n_w$ ,  $E(n)$  is the recording. Based on the history, the recording is used to identify the new activity that was initialized in the instructor stage. This instructor holds the classification model’s precision and the least positive failure. From this stage, the bots’ recognition is processed, providing error detection in new activity. The recording is periodically processed for the new activity performed in this process. The forthcoming process is used for the classification model that describes the classes and labels for the two neural classifications mentioned above, such as precision and least positive failure. The following equation is processed for the multi-classification in this work.

$$\alpha(M) = \underbrace{(n_w + o') * \sum_R(b_t + n_i)}_{Class} + \underbrace{(m_i - P_h) + \prod_{A+b_t}(o' + m_p)}_{Label} \tag{8}$$

The multiple classifications are introduced in the above equation, where it opts for the classes and label, and it is described as  $\alpha(M)$ . The first derivation is the classes that state to find the appropriate operation of the bot’s performances. Based on this, recognition is performed to plan the task among the n-layers in the neurons. The second is the label classification used to fix the bots’ control so they can start the operation with certain controls. This approach perceives this detection for the robot’s operation and control detection. Thus, multiple classifications are used in this deep neural network for precision and accuracy. Based on these classes and labels, the classification in neural networks is performed reliably. From this stage, the analysis is carried out for the instructor and formulated below.

$$Y = \sum_{i_p}^{b_t} (n_w + R) * (P_r + L_f) + \left(\frac{R + A}{m_p - P_h}\right) \tag{9}$$

The instructor analysis is carried out based on the input acquired from the classification module. This method indicates the precision and least positive failure, and they are symbolized as  $P_r$  and  $L_f$ . In this process, the recognition maps the history from which the new activities are initiated. This starting stage indicates recognition based on the mapping of the history and provides reliable computing. The deep neural network for classification is presented in Figure 4.

The classification using deep neural networks is a two-fold process defined by  $m_p = 1$  and  $m_p = 0$ . The mapping case is to validate the understandability at any interval of M such that A satisfies  $\alpha = high$  case (output 1). The  $n_w$  case is validated for the instructions pursued and, thereby, to verify if  $\alpha(M)$  is feasible. The least feasible solutions are the  $\alpha = low \leq \alpha < \alpha = high$  in deciding the outputs. Based on  $\alpha = high$  and unknown, the  $i_p = \sum(G', S', E')$  is recognized to improve the bots  $o'$  response. The instruction-based validation is expelled for  $e_r$  under the consecutive training of the possibilities (as in Figure 3) for  $\alpha = high$  achievement (Refer to Figure 4). Here, this methodology indicates the bots’ control and assistance under the classification based on the training. Here, the instructor holds the classification module’s precise and least positive failure. In this case, the training phase is based on history and new activities and is equated in the equation below.

$$T_h = (P_h + \alpha) + n_l - m_i * (U_d + o') \tag{10}$$

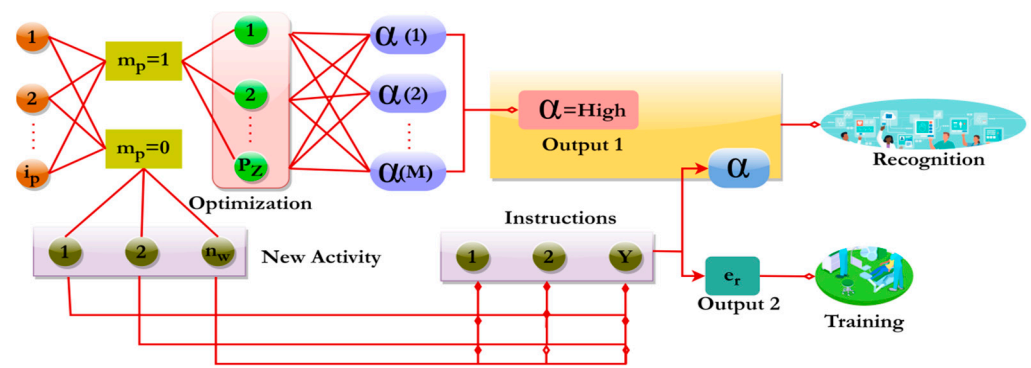


Figure 4. Deep Neural Network Representation for Classification.

The training phase from the history and new activities states the task planning it embeds with the computation time. This stage indicates the recognition of the bot’s operation and control of the task in AI; the understandability is  $U_d$ . The deep neural classification model is proposed for the optimization result in this training phase, and it is described as  $T_h$ . Since this indicates the precise and least positive failure in detecting the error in bot operation. Thus, this processing step involves optimally understanding the bot’s operation and control. This comparable validation between detection and unrecognizable input is processed along with the understandability observed in the formulations below.

$$C_b = (P_h - m_p) + b_t * \alpha + \prod_{T_r} [(R + A) * D] + U_d \tag{11}$$

where,

$$U_d = ((D + b_t) * R) + m_p - P_h + T_h \tag{12}$$

Comparability and understandability rely on this paper for the efficient result under the detection of error, and it is represented as  $C_b$ . This structural formulation provides the classification module that indicates the mapping between task planning and the completion time. In this case, the new activities differentiate between the recognition and unrecognition approaches. This step indicates a better understandability link between the instructors and the new activities. Equation (6) defines the neural classification where the instructors are responsible for comparing and understanding the bots. In this case, the following equation is used for the recognition based on the new activities derived below.

$$R(n_w) = (A + i_p) * b_t(o') + m_p - P_h(T_h) - e_r + D \tag{13}$$

In this derivation, the new activities are recognized by acquiring input from the bots and following up on the controls. Based on this training phase, it deploys the mapping with the history and provides the recognition. The new activities detected are recorded and provide consistent recognition. If the recognition is imperfect, then the training phase boots to train the particular layer obtained from the new activities. It integrates the training from history for better recognition in the next neural layer. Thus, the bots operate on the AI, and because of this understandability, errors are reduced. Therefore, the proposed technique is validated using assistance response, time, and error factors.

### 3.2. Data Analysis

The data-based analysis of this article relies on the THU-HRIA dataset provided in (<https://ieee-dataport.org/documents/thu-hria-dataset-human-robot-interactive-action-dataset-perspective-service-robot#files>, accessed on 15 May 2024). This dataset is generated using eight different actions of people interacting with an activity monitoring and guidance robot. The instructions are classified based on their inputs (actions) observed between 1.5 m and 3 m range. The input duration is a minimum of 5 s to a maximum of 20 s, requiring 576 training inputs. Based on this input, let the actions be 8 (as given), for which  $n_l$  is analyzed in Table 1.

**Table 1.**  $n_l$  Analysis for Action Inputs.

Action	A = 2	A = 4	A = 6	A = 8	$n_l$
Waving	0.17	0.41	0.465	0.85	0.89
Calling	0.18	0.32	0.65	0.9	0.85
Beckon	0.21	0.42	0.52	0.95	0.82
Move Backwards	0.25	0.38	0.7	0.89	0.85
Crossing	0.23	0.45	0.65	0.92	0.81
Sliding	0.32	0.48	0.58	0.89	0.74
Speech	0.38	0.5	0.49	0.99	0.55
Smiling	0.32	0.45	0.59	0.95	0.59

Table 1 presents the  $n_l$  for the varying  $A$  and actions incorporated from the dataset. The initial  $i_p$  is formulated by considering the  $O'$  and  $P_h$  (if any) for which the planning is performed. The previous completion time is validated in this planning process to improve the task processing/optimization and generate free-flow control. Under the  $n_l$ , the high input ranges experience better optimization from which  $O'$  is levied. This is random for different  $\alpha$  levels that are analyzed further. Following the above tabulation, the  $\alpha$  with  $e_r$  for the 8 actions/activities are analyzed with the least and maximum variations in Table 2.

**Table 2.**  $\alpha$  and  $e_r$  Analysis.

	Waving		Calling		Beckon		Moving Backwards		Crossing		Sliding		Speech		Smiling	
	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$	$\alpha$	$e_r$
Waving	1	0	0.8	±0.12	0.64	±0.101	0.58	±0.06	0.7	±0.08	0.85	±0.089	0.78	±0.97	0.74	±0.2125
Calling	0.49	±0.06	1	0	0.71	±0.120	0.62	±0.087	0.8	±0.108	0.92	±0.096	0.85	±0.10	0.85	±0.085
Beckon	0.47	±0.084	0.85	±0.089	1	0	0.71	±0.098	0.6	±0.098	0.74	±0.098	0.91	±0.087	0.69	±0.087
Move Backwards	0.51	±0.097	0.52	±0.07	0.85	±0.089	1	0	0.54	±0.085	0.85	±0.123	0.48	±0.074	0.74	±0.134
Crossing	0.61	±0.092	0.49	±0.087	0.65	±0.089	0.85	±0.13	1	0	0.6	±0.124	0.91	±0.10	0.85	±0.13
Sliding	0.58	±0.84	0.52	±0.089	0.74	±0.096	0.49	±0.078	0.74	±0.085	1	0	0.85	±0.135	0.91	±0.14
Speech	0.48	±0.74	0.61	0.078	0.65	±0.106	0.48	±0.87	0.59	±0.139	0.9	±0.089	1	0	0.87	±0.098
Smiling	0.51	±0.65	0.74	±0.69	0.71	±0.14	0.87	±0.121	0.78	±0.14	0.78	±0.098	0.74	±0.139	1	0

The least and maximum variation are indicated by the  $I$  symbol in the above tabulation. Based on the  $A$  value, the coincidence of  $i_p = A$  achieves  $\alpha = 1$  and  $e_r = 0$ . The  $n_l$  optimization throughout  $T_Z$  and  $m_p$  improves the  $\alpha$  by classifying (excluding)  $m_p = 0$  cases. If the cases are excluded, then  $Y$  based validations are prompt in achieving high inputs. Therefore, the response to these inputs is high enough to reduce  $e_r$ . This tabulation is presented in Table 2, followed by the  $L_f$  under different conditions analyzed. This computation is referenced in Table 3.

**Table 3.**  $L_f$  Analyses.

Condition	Case	$L_f$
$\alpha = D - P_S$	Precision	

Table 3. Cont.

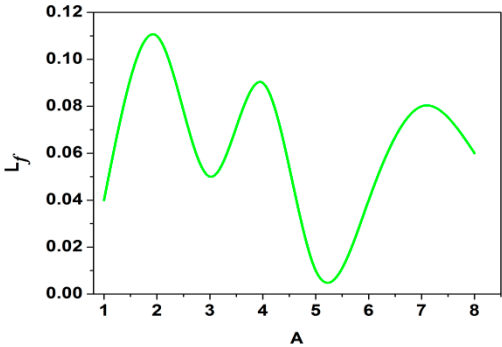
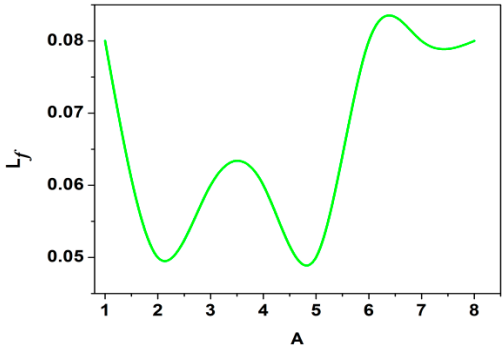
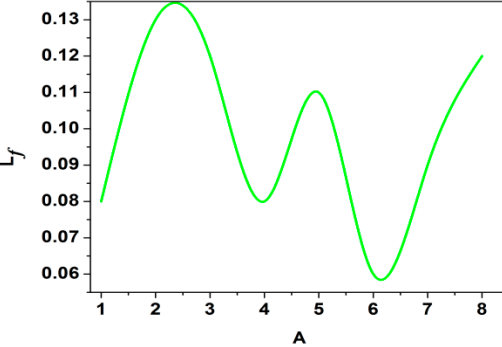
Condition	Case	$L_f$
$\alpha = D - P_S$	Error	
	Precision	
$\alpha = m_i$	Error	

Table 3 above, the  $L_f$  for different  $\alpha$  conditions are analyzed. This involves the precision and error scenario under the  $n_l$  optimization and learning  $T_z$ . This  $L_f$  is the variant that identifies different  $Y$  and activities for which precision (of response) is to be improved. Hence, in this case, the need for  $\alpha(M)$  increases the chances of error reduction with confined  $L_f$ . Now, with the  $L_f$  identified,  $R(N_w)$  is tabulated in Table 4.

In the above Table 4, the possibilities  $A = i_p, A \neq i_p, L_f = 1$  are analyzed. In these conditions,  $A = i_p$  and  $L_f = 0$  are the maximum  $R(N_w)$  generation cases. The learning process migrates  $L_f = 1$  to  $L_f = 0$  and  $A \neq i_p$  to  $A = i_p$  such that the response is swift. Based on  $T_z$  and  $P_z$  further optimization and learning iterations are defined. As the above factors are high, the  $R(N_w)$  increases provide new chances for input. These measures likely represent the probabilities or occurrences of an action  $A$  being either identical to or different from a reference or predicted action  $i_p$ . This could be useful for comparing how well the robot's actions match or diverge from expected or desired actions, which is critical in evaluating the accuracy and reliability of the robot's responses. A thorough assessment is a key component of the framework used to compare the neural classifier with the traditional classifier. This evaluation helps identify each of the strengths

and drawbacks. The two classifiers are trained and fine-tuned using a broad dataset, including voice commands and gestures. Input analysis efficiency, accuracy of assistance response, error detection rate, response time, and input detection accuracy are some of the performance indicators that are rigorously evaluated. This study evaluates these measures using statistical tests and visualizations and finds that the neural classifier is always more accurate, efficient, and responsive than the traditional classifier. Furthermore, user feedback and real-world testing demonstrate that the neural classifier offers a superior and more fulfilling user experience. After reviewing all the data, it's clear that the neural classifier is the way to improve human-robot interaction in this specific scenario.

**Table 4.**  $R(N_w)$  Analysis.

Action	$A = i_p$	$A \neq i_p$	$L_f = 1$	$L_f = 0$	$R(N_w)$
Waving	0.43	0.07	0.03	0.87	0.74
Calling	0.52	0.085	0.048	0.85	0.81
Beckon	0.63	0.097	0.052	0.74	0.94
Move Backwards	0.74	0.14	0.087	0.65	0.99
Crossing	0.74	0.25	0.091	0.81	0.95
Sliding	0.85	0.32	0.2	0.64	0.58
Speech	0.91	0.41	0.44	0.56	0.64
Smiling	0.85	0.38	0.31	0.61	0.84

#### 4. Results and Discussions

The discussion is presented as a comparative analysis using input detection, analyses, assistance response, error detection, and response time metrics. In this comparative analysis, the user inputs varied from 10 to 140, and the interaction time varied from 2 to 24 min. In this comparative analysis, the proposed technique is accompanied by the LQT + RLS [31], JISA [33], and HRC-DMP [19] methods discussed in the related works section. Inputs and durations vary greatly because users' preferences and technological expertise influence their interactions with the robot. The amount of time spent interacting is greatly impacted by the difficulty of the activities and queries. Processing and response generation for more complicated interactions take longer than simpler tasks. All the analysis of interaction time to understand interaction time helps optimize the robot's responsiveness, ensuring that it can interact with humans in a timely and efficient manner. Quick and accurate responses are essential for a positive user experience. Table 5 shows the experimental setup.

**Table 5.** Experimental Setup.

Component	Details
Sensors	LIDAR, depth cameras, touch sensors, microphones
Processing Unit	Intel Core i7 processor with 16 GB RAM for real-time processing
Motion Capture System	Vicon motion capture system for tracking user gestures and movements
Operating System	Robot OS (ROS) running on Ubuntu 18.04 LTS

##### 4.1. Input Detection

The input detection for the proposed work increases for the varying robots that perform the operation and control in AI. Here, the detection process indicates the history of where the deep neural network is used in this work. The precise and least positive failures are classified based on the input that has been acquired. In this stage, the new activities are developed from the classification model by examining the comparison and its understandability. Based on these two factors, input detection is performed by AI. Thus, the computation step indicates the recognition of the bot's action in AI, and based on this, classification is performed. From the classification model, the history is acquired where it

indicates the training. The training is based on the previous state of the process, and based on this, the recognition is carried out accurately. This processing step recognizes the new and the history, providing the result. Thus, the input detection is measured for human activities and robot controls, and it is equated as  $\left\{ \left[ (G' + S' + E') * \frac{i_p}{b_t} \right] \right\}$  (Figure 5).

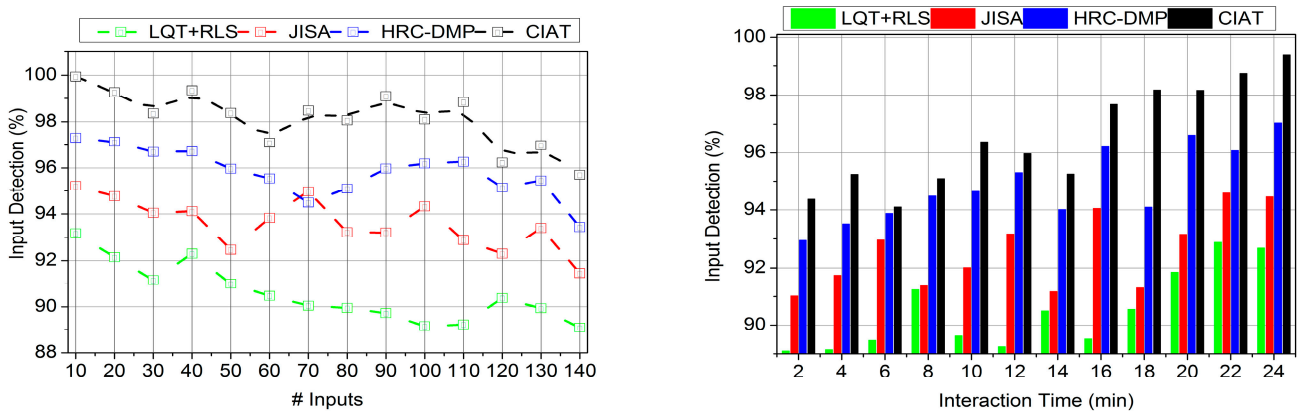


Figure 5. Input Detection.

4.2. Input Analyses

In Figure 6, the input analysis for the proposed work is enhanced for the different training phases in history. This includes the training history of the robot action, and based on this, the instructors are used to perform the recognition. The recognition is followed up with the mapping method, which includes the control and operation of the robot. Here, understandability and comparison are used to deploy the new activities, and based on this planning task, computation time is associated with this bot’s action. The task completion within the required period is associated with finding the error; from this, history is associated with the mapping. The mapping is used to determine whether the new activities match the past ones; from this, the completion time is associated with the understandability. Understandability is based on recognition developed for multiple classifications, including classes and labels. In this analysis, the training is used to find the new activities, and based on this, the instructors are developed, and it is represented as

$$\left( 1 + \frac{R+A}{\sum p_h (n_l + m_i)} \right).$$

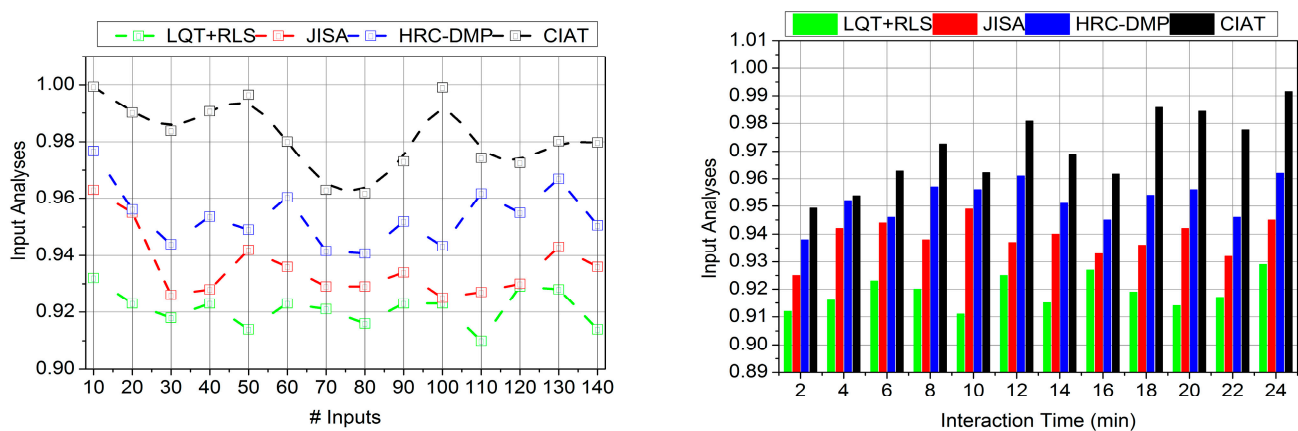


Figure 6. Input Analyses.

4.3. Assistance Response

The assistance response is higher if the correct planning is executed in this work and better training optimization is provided. Here, the training phase includes the history

of the instructors associated with the human-assisting robots. This recognition process indicates the precise and least positive failure that deploys the new activities, and based on this, AI optimization is carried out appropriately. In this manner, assistance response is improved for training optimization, and it is formulated as  $\left[ \left( \frac{n_l + m_i}{D + b_t} \right) + \left( \frac{R/P_h}{m_p - b_t} \right) \right]$ . The recognition is carried out from the history, and the task planning is used to perform better bot operations. Here, task planning and computation time are used to estimate the better classification among the precise and least positive failures. In this stage, acquiring the input for the human assisting response is equated as  $(i_p + b_t) * \left( \frac{(T_r + o')}{\sum_R (n_l + m_i)} \right)$ . Thus, the assistance response for the proposed work increases for task completion in the mentioned period. By deploying human activity for the new task, the classification and the response are improved accurately (Figure 7).

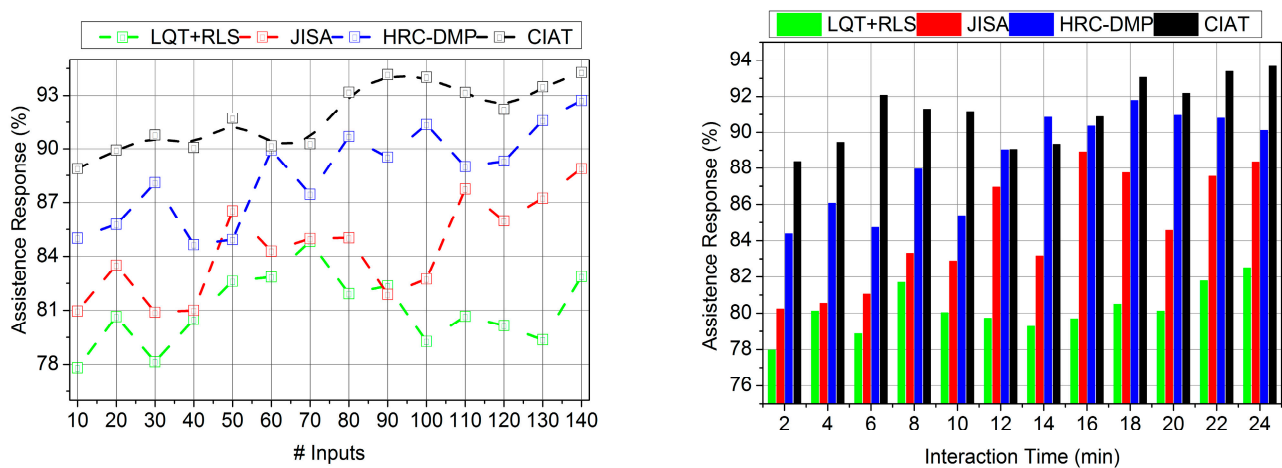


Figure 7. Assistance Response.

#### 4.4. Error Detection

In Figure 8, the error detection is reduced if the recording of new activities is associated with recognizing the bot’s operations and control. This is based on the processing history and involves the classification model. The classification model indicates the comparable validation between the detection and unrecognition of the particular task. In this category, task planning provides better multiple classifications distributed as class and label. From this classification, it invokes the instructors by fetching the input from the bot control. This approach reduces the error detection stage if the precise and least positive failure is associated with the new activities. Here, it is associated with the planning task and classification model, and from this, the history is used to map the present process. Thus, the validation process is used to optimize AI by using a deep neural network. The completion time for the assigned task is used to produce a better output with less error, and it is represented as  $((D + b_t) * R) + m_p$ .

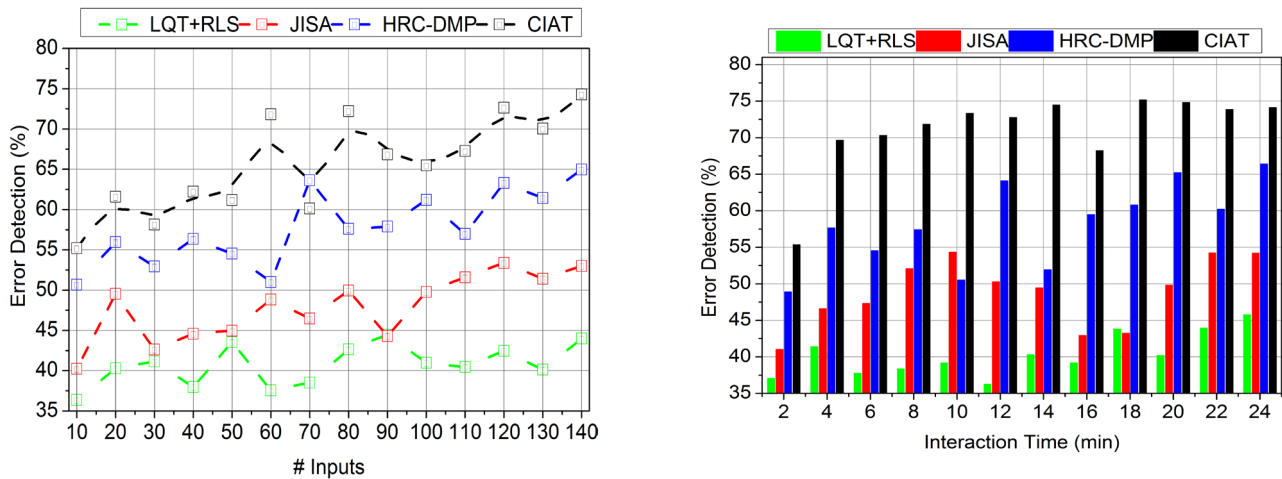


Figure 8. Error Detection.

#### 4.5. Response Time

The response time decreases for the human-assisting robots that deploy the training optimization. The recognition of this activity indicates the planning task and completion time. In this case, the understandability of robot control and operations is used to increase the precision of the classification model. By deploying this optimization in AI, the response time from the robot is decreased. In this manner, the least positive failure is addressed by the classification model, and the necessary steps are taken to avoid it in the forthcoming process. The human-assisting robots are run by recognizing the error factor from the instructors, where the new activities are detected from the history. The mapping is performed with the current method and provides the understandability of the robot, and it is formulated as  $\sum_A (b_t + R) * m_p - P_h + (R * b_t(o'))$ . This processing step indicates the recognition stage of bots and control optimized in this neural network. Thus, the bot's operation is performed by recognizing the error from the recorded new activity and ensuring the response time is reduced (Figure 9). The findings of the above discussion are summarized in Table 5 (# Inputs) and Table 6 (Interaction Time).

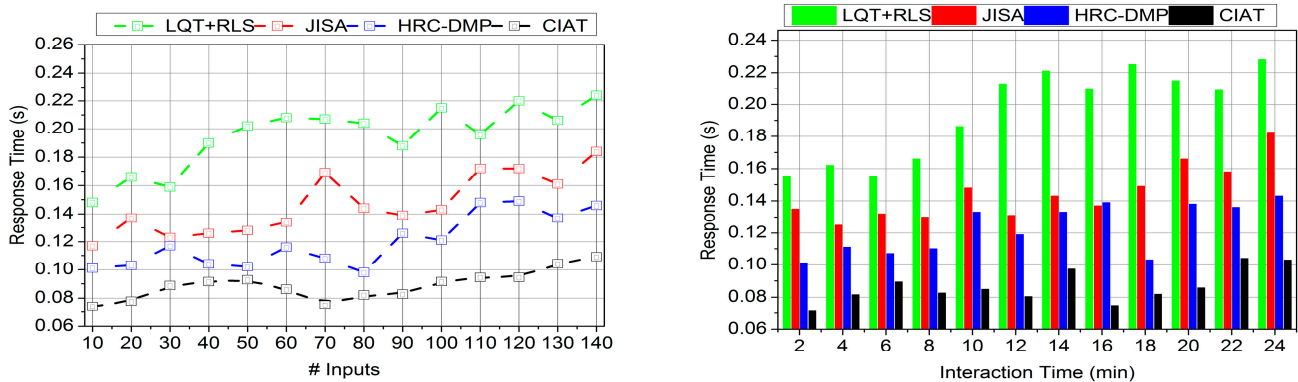


Figure 9. Response Time.

**Table 6.** Findings for # Inputs.

Metrics	LQT + RLS	JISA	HRC-DMP	CIAT	Findings
Input Detection (%)	89.09	91.47	93.44	95.698	8.73% High
Input Analyses	0.914	0.936	0.9205	0.9798	11.26% High
Assistance Response (%)	82.87	88.88	92.73	94.274	12.23% High
Error Detection (%)	43.98	52.98	64.97	74.277	10.15% High
Response Time (s)	0.224	0.184	0.146	0.1091	6.81% Less

As shown in Tables 6 and 7, the input detection measure evaluates the accuracy with which the system recognizes and processes user inputs from different interactions. It is calculated by dividing the total inputs by the proportion of accurate detections. The input detection rate of a system would be 92% if, out of 100 input directions, it could successfully identify 92% of them. This statistic directly impacts interaction and efficiency, which is vital for comprehending the system's ability to discern user commands or gestures. Metrics for input analyses measure how well the system handles and understands the detected inputs. It measures the system's intelligence in interpreting data and drawing conclusions. The measure is often shown as a normalized value, where larger values imply greater performance. The range of possible values is 0 to 1. A rating of 0.95, for example, indicates that the system's input analysis is accurate and efficient. This is crucial to ensure the system gets the user's intent right and reacts properly. One measure of assistance response is the proportion of suitable and accurate responses given by the system in response to the inputs that have been examined. It shows how closely the results produced by the system match the requirements or wishes of the user. For example, the system's help response rate would be 85% if it correctly responded to 85 out of 100 user inputs. This indicator is essential for measuring how well the system helps users and ensuring the answers are relevant and valuable. The error detection metric measures the system's capacity to identify mistakes in the incoming data or its own responses. It is the proportion of mistakes that the system identified as a percentage of all errors accurately. For example, the system's error detection rate would be 70% if it correctly found 70% of the faults. Since, it aids in finding and fixing errors, boosting overall system performance, and gaining user confidence, high error detection is critical for keeping systems reliable. This statistic measures how long it takes for the system to respond to a user's input, measured in seconds. It encompasses the time it takes for the system to react after receiving an input. A response time of 0.2 s indicates, for example, that the system requires 0.2 s to reply to a command given by the user. Ensuring users get feedback and help quickly is essential for an easy and smooth reaction experience, so response times should be kept to a minimum.

**Table 7.** Findings for Interaction Time.

Metrics	LQT + RLS	JISA	HRC-DMP	CIAT	Findings
Input Detection (%)	92.68	94.45	97.04	99.374	9.3% High
Input Analyses	0.929	0.945	0.962	0.9915	9.23% High
Assistance Response (%)	82.49	88.34	90.11	93.668	13.38% High
Error Detection (%)	45.76	54.2	66.4	74.102	9.32% High
Response Time (s)	0.228	0.182	0.143	0.1029	7.38% Less

## 5. Conclusions

Robotic systems for human assistance in public and personal applications are incomparable, so a prompt assessment technique was introduced. The technique introduced in this manuscript was intended to improve the robot system's understandability regardless of different input forms. The thorough classification and assessment of the input optimized the cooperative control for human assistance with the highest possible instruction detection. The inputs are analyzed using neural classification to identify the highest possible

instruction input with fewer assessments. The robot system was trained using previous instructions and assistance activities pursued with human input histories. The problem of unrecognizable inputs was a time-consuming assessment in this technique that is differentiated using the neural classifier. Thus, precision input and comparable validations to reduce understandability were pursued in this technique. Modern neural classifiers and real-time learning methods have the potential to enhance human-robot interactions greatly. The result is more natural, intuitive, and effective interaction, making robots more useful and acceptable for daily tasks. Robots may be made to work consistently in constantly changing and unexpected situations if this model builds systems to identify and handle errors well and use adaptive learning in real-time. This enhances the reliability and trustworthiness of robotic systems in critical applications. In the medical field, robots can keep an eye on patients and help the elderly; in the classroom, they can provide individualized lessons and aid students with special needs; in retail and public service, they can improve customer service; at home, they may sweep, vacuum, and communicate with residents; at work, they can increase efficiency; and in entertainment, they can participate in interactive games and fitness programs. This multi-disciplinary study advances HRI technologies by combining computer vision, natural language processing (NLP), and cognitive science; its solutions improve people's lives and make various operations more efficient. As a result, the proposed technique was found to reduce response time by 6.81%, improve input detection by 8.73%, and provide assistance by 12.23% under varying inputs. Future work will utilize edge computing to process data locally on the robot, reducing latency, improving responsiveness, and implementing distributed systems to handle computationally intensive tasks.

**Author Contributions:** Methodology, M.A., K.K., G.A., P.M., M.D.A. and A.A.; Software, M.A., K.K., G.A., P.M., M.D.A. and A.A.; Validation, M.A., K.K., G.A., P.M., M.D.A. and A.A.; Formal analysis, M.A., K.K., G.A., P.M., M.D.A. and A.A.; Investigation, M.A., K.K., G.A., P.M., M.D.A. and A.A.; Resources, P.M.; Data curation, M.A., G.A., M.D.A. and A.A.; Writing—original draft, M.A., K.K., G.A., P.M., M.D.A. and A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Deanship of Graduate Studies and Scientific Research at Jouf University under grant No. (DGSSR-2024-02-01087).

**Data Availability Statement:** The data will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Olugbade, T.; He, L.; Maiolino, P.; Heylen, D.; Bianchi-Berthouze, N. Touch Technology in Affective Human–, Robot–, and Virtual–Human Interactions: A Survey. *Proc. IEEE* **2023**, *111*, 1333–1354. [[CrossRef](#)]
2. Zahedi, F.; Arnold, J.; Phillips, C.; Lee, H. Variable damping control for phri: Considering stability, agility, and human effort in controlling human interactive robots. *IEEE Trans. Hum.-Mach. Syst.* **2021**, *51*, 504–513. [[CrossRef](#)]
3. Ding, B.; Li, Y.; Miah, S.; Liu, W. Customer acceptance of frontline social robots—Human-robot interaction as boundary condition. *Technol. Forecast. Soc. Chang.* **2024**, *199*, 123035. [[CrossRef](#)]
4. Chou, S.Y.; Barron, K.; Ramser, C. Paradox in the making: Toward a theory of utility maximization in human-commercial robot interactions. *J. Organ. Chang. Manag.* **2023**, *36*, 1144–1162. [[CrossRef](#)]
5. Fiorini, L.; Coviello, L.; Sorrentino, A.; Sancarolo, D.; Ciccone, F.; D’Onofrio, G.; Cavallo, F. User Profiling to Enhance Clinical Assessment and Human–Robot Interaction: A Feasibility Study. *Int. J. Soc. Robot.* **2023**, *15*, 501–516. [[CrossRef](#)] [[PubMed](#)]
6. Xing, H.; Torabi, A.; Ding, L.; Gao, H.; Deng, Z.; Mushahwar, V.K.; Tavakoli, M. An admittance-controlled wheeled mobile manipulator for mobility assistance: Human–robot interaction estimation and redundancy resolution for enhanced force exertion ability. *Mechatronics* **2021**, *74*, 102497. [[CrossRef](#)]
7. Tolba, A.; Al-Makhadmeh, Z. Modular interactive computation scheme for the internet of things assisted robotic services. *Swarm Evol. Comput.* **2022**, *70*, 101043. [[CrossRef](#)]
8. Fardeau, E.; Senghor, A.S.; Racine, E. The Impact of Socially Assistive Robots on Human Flourishing in the Context of Dementia: A Scoping Review. *Int. J. Soc. Robot.* **2023**, *15*, 1025–1075. [[CrossRef](#)]
9. Nocentini, O.; Kim, J.; Bashir, Z.M.; Cavallo, F. Learning-based control approaches for service robots on cloth manipulation and dressing assistance: A comprehensive review. *J. NeuroEng. Rehabil.* **2022**, *19*, 117. [[CrossRef](#)]

10. Patrício, M.L.; Jamshidnejad, A. Dynamic mathematical models of theory of mind for socially assistive robots. *IEEE Access* **2023**, *11*, 103956–103975. [[CrossRef](#)]
11. Erickson, Z.; Clever, H.M.; Gangaram, V.; Xing, E.; Turk, G.; Liu, C.K.; Kemp, C.C. Characterizing Multi-dimensional Capacitive Servoing for Physical Human–Robot Interaction. *IEEE Trans. Robot.* **2022**, *39*, 357–372. [[CrossRef](#)]
12. Zhang, Z.; Ji, Y.; Tang, D.; Chen, J.; Liu, C. Enabling collaborative assembly between humans and robots using a digital twin system. *Robot. Comput.-Integr. Manuf.* **2024**, *86*, 102691. [[CrossRef](#)]
13. Liu, S.; Fantoni, I.; Chriette, A. Decentralized control and state estimation of a flying parallel robot interacting with the environment. *Control Eng. Pract.* **2024**, *144*, 105817. [[CrossRef](#)]
14. Wojtak, W.; Ferreira, F.; Louro, L.; Bicho, E.; Erlhagen, W. Adaptive timing in a dynamic field architecture for natural human–robot interactions. *Cogn. Syst. Res.* **2023**, *82*, 101148. [[CrossRef](#)]
15. Wang, Y.X.; Meng, Q.H.; Li, Y.K.; Hou, H.R. Touch-text answer for human-robot interaction via supervised adversarial learning. *Expert Syst. Appl.* **2024**, *242*, 122738. [[CrossRef](#)]
16. Liu, C.; Zhang, Z.; Tang, D.; Nie, Q.; Zhang, L.; Song, J. A mixed perception-based human-robot collaborative maintenance approach driven by augmented reality and online deep reinforcement learning. *Robot. Comput.-Integr. Manuf.* **2023**, *83*, 102568. [[CrossRef](#)]
17. Odesanmi, G.A.; Wang, Q.; Mai, J. Skill learning framework for human–robot interaction and manipulation tasks. *Robot. Comput.-Integr. Manuf.* **2023**, *79*, 102444. [[CrossRef](#)]
18. Sheron, P.F.; Sridhar, K.P.; Baskar, S.; Shakeel, P.M. Projection-dependent input processing for 3D object recognition in human robot interaction systems. *Image Vis. Comput.* **2021**, *106*, 104089. [[CrossRef](#)]
19. Liao, Z.; Lorenzini, M.; Leonori, M.; Zhao, F.; Jiang, G.; Ajoudani, A. An Ergo-Interactive Framework for Human-Robot Collaboration Via Learning From Demonstration. *IEEE Robot. Autom. Lett.* **2023**, *9*, 359–366. [[CrossRef](#)]
20. Hindemith, L.; Bruns, O.; Noller, A.M.; Hemion, N.; Schneider, S.; Vollmer, A.L. Interactive robot task learning: Human teaching proficiency with different feedback approaches. *IEEE Trans. Cogn. Dev. Syst.* **2022**, *15*, 1938–1947. [[CrossRef](#)]
21. Qian, K.; Xu, X.; Liu, H.; Bai, J.; Luo, S. Environment-adaptive learning from demonstration for proactive assistance in human–robot collaborative tasks. *Robot. Auton. Syst.* **2022**, *151*, 104046. [[CrossRef](#)]
22. Di Marino, C.; Rega, A.; Pasquariello, A.; Fruggiero, F.; Vitolo, F.; Patalano, S. An interactive graph-based tool to support the designing of human–robot collaborative workplaces. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2023**, 1–16. [[CrossRef](#)]
23. Burks, L.; Ray, H.M.; McGinley, J.; Vunnam, S.; Ahmed, N. HARPS: An Online POMDP Framework for Human-Assisted Robotic Planning and Sensing. *IEEE Trans. Robot.* **2023**, *39*, 3024–3042. [[CrossRef](#)]
24. Muramatsu, H.; Itaguchi, Y.; Katsura, S. Involuntary Stabilization in Discrete-Event Physical Human–Robot Interaction. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 576–587. [[CrossRef](#)]
25. Fu, C.; Liu, C.; Ishi, C.T.; Yoshikawa, Y.; Iio, T.; Ishiguro, H. Using an android robot to improve social connectedness by sharing recent experiences of group members in human-robot conversations. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6670–6677. [[CrossRef](#)]
26. Rückert, P.; Wallmeier, H.; Tracht, K. Biofeedback for human-robot interaction in the context of collaborative assembly. *Procedia CIRP* **2023**, *118*, 952–957. [[CrossRef](#)]
27. Li, J.; Lu, L.; Zhao, L.; Wang, C.; Li, J. An integrated approach for robotic Sit-To-Stand assistance: Control framework design and human intention recognition. *Control Eng. Pract.* **2021**, *107*, 104680. [[CrossRef](#)]
28. Yu, X.; Zhang, X.; Xu, C.; Ou, L. Human–robot collaborative interaction with human perception and action recognition. *Neurocomputing* **2024**, *563*, 126827. [[CrossRef](#)]
29. Zhou, H.; Yang, G.; Wang, B.; Li, X.; Wang, R.; Huang, X.; Wang, X.V. An attention-based deep learning approach for inertial motion recognition and estimation in human-robot collaboration. *J. Manuf. Syst.* **2023**, *67*, 97–110. [[CrossRef](#)]
30. Zhang, X.; Wang, Y.; Li, C.; Fahmy, A.; Sienz, J. Innovative multi-dimensional learning algorithm and experiment design for human-robot cooperation. *Appl. Math. Model.* **2024**, *127*, 730–751. [[CrossRef](#)]
31. Lippi, M.; Marino, A. Human multi-robot physical interaction: A distributed framework. *J. Intell. Robot. Syst.* **2021**, *101*, 35. [[CrossRef](#)]
32. Liao, Y.Y.; Ryu, K. Genetic algorithm-based task allocation in multiple modes of human–robot collaboration systems with two cobots. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 7291–7309. [[CrossRef](#)]
33. Sidaoui, A.; Daher, N.; Asmar, D. Human-robot interaction via a joint-initiative supervised autonomy (jisa) framework. *J. Intell. Robot. Syst.* **2022**, *104*, 51. [[CrossRef](#)]
34. Ince, G.; Yorganci, R.; Ozkul, A.; Duman, T.B.; Köse, H. An audiovisual interface-based drumming system for multimodal human–robot interaction. *J. Multimodal User Interfaces* **2021**, *15*, 413–428. [[CrossRef](#)]
35. Jarrah, M.I.M.; Jaya, A.S.M.; Azam, M.A.; Alqattan, Z.N.; Muhamad, M.R.; Abdullah, R. Application of bat algorithm in carbon nanotubes growing process parameters optimization. In *Intelligent and Interactive Computing: Proceedings of IIC 2018*; Springer: Singapore, 2019; pp. 179–192.
36. Shorman, S.; Jarrah, M.; Alsayed, A.R. The Websites Technology for Arabic Language Learning Through COVID-19 Pandemic. In *Future of Organizations and Work After the 4th Industrial Revolution: The Role of Artificial Intelligence, Big Data, Automation, and Robotics*; Springer International Publishing: Cham, Switzerland, 2022; pp. 327–340.

37. Saren, S.; Mukhopadhyay, A.; Ghose, D.; Biswas, P. Comparing alternative modalities in the context of multimodal human–robot interaction. *J. Multimodal User Interfaces* **2024**, *18*, 69–85. [[CrossRef](#)]
38. Wang, T.; Zheng, P.; Li, S.; Wang, L. Multimodal Human–Robot Interaction for Human-Centric Smart Manufacturing: A Survey. *Adv. Intell. Syst.* **2024**, *6*, 2300359. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.