AN ABSTRACT OF THE DISSERTATION OF

Prakash Baskaran for the degree of Doctor of Philosophy in Robotics presented on December 12, 2023.

Title: Multi-Dimensional Task Recognition for Human-Robot Teaming

Abstract approved: _____

Julie A. Adams

Human-robot teams involve humans and robots collaborating to achieve tasks under various environmental conditions. Successful teaming requires robots to adapt autonomously in real-time to a human teammate's state. An important element of such adaptation is the ability for the robot to infer the tasks performed by their human teammates. Human-robot teams often perform a wide variety of tasks, involving multiple activity components, and may even perform two or more tasks concurrently. A robot's ability to recognize the human's composite tasks that occur concurrently is a key requirement for realizing successful collaboration. Existing task recognition algorithms are not viable for human-robot teams, as they only detect tasks from a subset of activity components and rarely detect concurrent, composite tasks. This dissertation developed a multi-dimensional task recognition algorithm capable of detecting concurrent, composite tasks across the cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile components by incorporating metrics that are sensitive, versatile, and suitable across human-robot teaming paradigms. The developed algorithm addresses a foundational problem of understanding an individual's task engagement state in human-robot teams operating in dynamic, unstructured environments.

Multi-Dimensional Task Recognition for Human-Robot Teaming

by

Prakash Baskaran

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 12, 2023
Commencement June 2024

Doctor of Philosophy dissertation of Prakash Baskaran presented on December 12, 2023.

APPROVED:

Major Professor, representing Robotics

Associate Dean of Graduate Studies for the College of Engineering

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Prakash Baskaran, Author

ACKNOWLEDGEMENTS

The first and foremost person I must thank for helping and supporting my Ph.D. journey is my wife, Dharani Thirumalaisamy. Thank you for being there every step of the way and always believing in me, especially on days when I failed to believe in myself. I love you more than you can possibly know, and I cannot imagine undergoing this journey without you being by my side.

Secondly, I want to express my gratitude to my advisor Dr. Julie A. Adams for being my pillar of support. The amount of professional growth I have attained under your mentorship is tremendous. Thank you for pushing me constantly to challenge myself, and molding me into the person I am today. I'm grateful for your guidance and support throughout my journey. My doctoral committee members were also imperative in the formation of this doctoral thesis. I want to thank each member and express my appreciation for the constructive criticism and feedback.

I also want to thank my current and former Human-Machine Teaming labmates at Oregon State University. A healthy lab culture is imperative to anyone pursuing doctoral studies. My labmates Joshua Bhagat Smith, Mark Robin Giolando, Neha Pusalkar, Dr. Jennifer Leaf, and Dr. Grace Diehl are some of the best people when it comes to creating such a culture. Thank you all for your mental, emotional, and technical support. My sincere thanks to Dr. Jamison Heard for mentoring me during my early Ph.D. days.

Lastly, I want to thank my parents, Baskaran Saravanamoorthy and Meenakshi Baskaran, for all the hardships they endured to ensure that I had the best platform to go and express myself. I dedicate this dissertation to you both.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

LIST OF TABLES (Continued)

Table Page

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

Table Page

LIST OF APPENDIX FIGURES

LIST OF APPENDIX FIGURES (Continued)

Figure Page

LIST OF APPENDIX FIGURES (Continued)

Figure Page

LIST OF APPENDIX TABLES

LIST OF VARIABLES

A^f The GNN fusion algorithm's adjacency matrix.

$A^f_{i,j}$ The pearson's correlation coefficients between the i^{th} and j^{th} components.

G^f The GNN fusion algorithm's graph.

$H^{(l)}$ The graph network's node feature at the l^{th} layer.

$W^{(l)}$ The learnable weight parameter at the l^{th} layer.

$\Phi(s_k)$ A preprocessing function that converts sensor readings s_k into a feature vector d.

α_l The activation function at the l^{th} layer.

μ Mean or average value.

σ Standard deviation.

$h(.)$ The graph convolution filter operator.

$s_k^{I/E}$ Sensor readings of a 6-channel IMU or 8-channel sEMG.

s_k Sensor readings that corresponds to a task a_k.

t_s The stride duration of a sliding window in seconds.

t_w sliding window duration in seconds.

w_i^k IMPRINT Pro workload values assigned to the i^{th} component when performing a k^{th} task.

x_t The list of atomic tasks across the seven components at time t.

x_{t_i} The atomic of the i^{th} component at time t.

y_t The list of concurrent, composite tasks at time t.

LIST OF ACRONYMS

ADL	Activities of Daily Living
CNN	Convolutional Neural Network
ECG	Electrocardiography
EEG	Electroencephalography
EOG	Electrooculography
GNN	Graphical Neural Network
HRV	Heart Rate Variability
Hz	Hertz
HRTs	Human-Robot Teams
IMPRINT	Improved Performance Research Integration Tool
IMU	Inertial Measurement Unit
LSTM	Long Short-Term Memory
MFCCs	Mel-Frequency Cepstral Coefficients
MATB-II	Multi-Attribute Task Battery-II
NL	Normal load
OL	Overload
RFID	Radio Frequency Identification
RF	Random Forest
ReLU	Rectified Linear Units
s	seconds
std. dev.	standard deviation
SVM	Support Vector Machine
sEMG	Surface-Electromyography
TCN	Temporal Convolutional Network
UL	Underload

Chapter 1: Introduction

Human-robot teams involve humans and robots collaborating to achieve tasks under various environmental conditions. Understanding human actions and their interactions with the physical world provides the robot with more context as to what type of assistance the human may need, which is crucial for enabling natural and successful teaming. Natural teaming requires robots to adapt autonomously to a human teammate's state. An important element of such adaptation is the robot's ability to infer the tasks performed by its human teammates. This dissertation developed and evaluated algorithms to identify tasks performed by human teammates in unstructured, dynamic environments using non-intrusive, non-vision wearable sensors.

Human-Robot Teams (HRTs) are often required to operate in dynamic, unstructured environments. Task identification requires sensors to determine the actions undertaken by human teammates. However, teams in unstructured environments cannot rely on sensors (e.g., motion capture, cameras) that are embedded in the environment. Moreover, the usage of image or video data may raise privacy concerns, and can be computationally expensive to process in real-time, especially in time-critical applications (e.g., disaster response domains). Using non-vision wearable sensors can permit identifying human teammates' tasks in real-time.

Tasks performed by HRTs can involve various activity components: cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile. Gross motor movements involve motions that displace the entire body (e.g., walking, running, and climbing stairs), or major portions of the body (e.g., bending the torso and swinging an arm), while fine-grained motor movements involve the motion of body extremities (e.g., using wrists and fingers for grasping and object manipulation). Tactile interaction involves motions that cause a sense of touch (e.g., mouse clicks, and keyboard strokes). HRT tasks are often composite tasks in that they aggregate multiple series of coordinated actions across various activity components. For instance, responding to a message over Walkie-Talkie aggregates an *auditory* component of listening to the information, a *cognitive* component of processing the information to decide if a response is required, a *fine-grained motor* component of picking up the Walkie-Talkie, a *tactile* component of pressing and holding the Walkie-

Talkie, and finally, a *speech* response.

Current state-of-the-art HRT task recognition using wearable sensors is limited to gross motor and some fine-grained motor tasks. Prior research identified visual, cognitive, and some auditory tasks using wearable sensors; however, none of those methods recognize composite tasks across all activity components. A robot's ability to recognize the human's composite tasks is a key requirement for realizing successful HRT collaboration. This dissertation developed a multi-dimensional task recognition algorithm that utilizes wearable sensors for identifying composite tasks, involving seven activity components, performed by humans in unstructured, dynamic environments.

Another limitation of current HRT task recognition algorithms is the inability to recognize concurrent, or overlapping tasks. Most existing algorithms assume that an individual only performs one activity at a time, which is not the case for many HRT scenarios, where the human may perform two or more tasks concurrently. Consider a disaster response search operation, where humans are working with multiple unmanned ground and aerial vehicles. A human supervisor located in the warm zone (i.e., a safe distance from the contaminant, but close to the area) may be commanding a ground vehicle, while monitoring the status of an aerial vehicle simultaneously. At each time step, tasks that involve direct human interaction are identified as foreground tasks, while all other active tasks are identified as background tasks. Detecting this task concurrency will allow robots to better adapt to the team's interactions, priorities, or appropriations, which will improve the team's overall collaboration and performance. This dissertation presents a concurrency detection method that augments the multi-dimensional task recognition algorithm in order to detect composite tasks that occur simultaneously.

This dissertation's primary focus is the development of a multi-dimensional task recognition algorithm to identify tasks performed by HRTs working in unstructured, dynamic environments. The algorithm detects concurrent, composite tasks across multiple activity components using wearable sensors. The developed algorithm is validated across multiple task environments to assess its viability across domains. Chapter 2 provides background information on task recognition, metrics incorporated for task recognition, and reviews the existing task recognition algorithms by activity component, and validates their composite and concurrency capabilities. Chapter 3 discusses the methodology for developing the multi-dimensional task recognition algorithm. Chapters 4 and 5 validate the algorithm's performance across two HRT domains, while Chapter 6 outlines the primary contributions and suggests key future research directions.

Chapter 2: Related Work

Task recognition involves classifying an activity performed by an individual from a set of domain relevant activities, or tasks [271]. Assume a user is performing a task, a_k, belonging to a task set A. There exists a sequence of sensor readings s_k that corresponds to the task a_k. The objective of task recognition is to identify a function f that predicts the task performed based on the sensor readings s_k, such that the discrepancy between the predicted task $\hat{a_k}$ and the ground truth a_k is minimized. f does not usually take s_k as input directly, as the sensor readings are often processed using a function Φ that converts the sensor reading s_k into a d-dimensional feature vector $\Phi(s_k) \doteq \mathbf{d} \in \mathbb{R}^d$ by extracting meaningful features [271]. The function f takes the feature vector $\mathbf{d}$ as input, and predicts the task $\hat{a_k}$. Human's individual differences can result in the same task being performed in multiple different ways or with differing completion times, which can lead to a number of feature vectors being mapped to a single task. Therefore, machine learning algorithms are widely adopted for learning the function f, instead of solving it deterministically [146].

Task recognition algorithms use machine learning to analyze the underlying patterns in the features x extracted from the sensor data for classifying the tasks. Machine learning can be broadly divided into two categories: *supervised learning* and *unsupervised learning*. Supervised learning requires training with labeled data that has inputs and corresponding target outputs, while unsupervised learning does not require labeled training data [222]. Supervised learning is more common for task recognition [47], because it requires the learning model to predict a class label as output [146].

HRTs often operate in dynamic, uncertain, and unstructured environments that do not enable the use of static environmentally embedded sensors. Therefore, wearable sensors are preferred for a broader set of domains due to ease of deployment in environments with more uncertainty and dynamic aspects, while environmentally embedded sensors are generally more suitable for structured environments with less uncertainty (e.g., control rooms). HRT tasks may encompass multiple activity components: cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile. Cognitive tasks require a person to process new information mentally, as well as recall or retrieve that information from memory [117, 133, 234, 281, 286]). Speech-reliant tasks are performed by a person using their voice

(e.g., communicating with a supervisor over the radio) [6, 49]), while auditory tasks involve sensing acoustic events in the environment (e.g., listening to an important announcement or emergency sounds) [91, 144, 159, 256]). Visual tasks use the eyes to perform tasks (e.g., identifying different objects, and reading) [30, 98, 99, 253]. Gross motor tasks involve physical movements that displace the entire body (e.g., walking, running, and climbing stairs), or major portions of the body (e.g., swinging an arm) [10, 43, 45, 105, 151]. Fine-grained motor tasks involve the motion of body extremities (e.g., using wrists and fingers for grasping and object manipulation) [58, 127, 142, 282]. Tactile tasks involve physical movements that cause a sense of touch (e.g., mouse clicks and keyboard strokes) [93, 150]. Most existing task recognition approaches focus primarily on detecting tasks involving physical movements (i.e., gross motor and fine-grained motor tasks). However, some tasks (e.g., reading or identifying a moving target) may involve little to no physical movement. Robots need a holistic understanding of the various activity components involved in the tasks in order to detect those tasks accurately.

2.1 Typical Task Recognition Categories

Tasks with similar activity components or characteristics can be grouped into a single task category based on the task environment, context, and task types. Seven common task categories exist in the literature.

Ambulatory tasks involve an individual's mobility (e.g., walking, running, sitting, lying down, or ascending and descending the stairs or similar structures or objects [47, 205]). Ambulatory tasks typically encompass the gross motor activity component only, and are atomic in nature. Other task categories (e.g., patient monitoring [196, 275], elder care [191], and fall detection [108]) exist; however, the tasks pertaining to these categories also detect gross motor tasks, similar to the ambulatory tasks. Therefore, all such tasks are categorized as ambulatory. Detecting ambulatory tasks can be considered a solved problem, as recognition solutions with accuracies over 90% exist (e.g., [43, 107, 110, 147, 209]).

Activities of Daily Living (ADL) collectively describes the basic set of tasks performed by individuals in their day-to-day life (e.g., eating, cooking, brushing teeth [34, 58, 66, 142, 171, 176, 184, 200, 216, 274]). The ADL tasks encompass the gross motor, fine-grained motor, tactile, and visual components. Depending on the tasks detected, a research result may involve all or a subset of the aforementioned activity components.

Office tasks are more sedentary, requiring standing or sitting at a table or computer

desktop. The tasks include: copying and pasting text, browsing the web, reading from a printed paper or computer monitor, typing, and taking handwritten notes [30, 98, 116, 174, 253]. The tasks performed typically encompass the fine-grained motor, tactile, visual, and cognitive activity components.

Industrial tasks have physical demands (e.g., carrying heavy objects, using a screwdriver, attaching and detaching assembly parts) performed by workers in a manufacturing setting [62, 92, 127, 130]. These tasks are composed primarily of gross motor, fine-grained motor, and tactile components.

Object manipulation tasks consist of articulated activities that enable individuals to physically interact with objects of different shapes and sizes. Grasp motions (e.g., ulnar pinch, tripod grasp, precision disk) enable individuals to hold and maneuver various objects and are common object manipulation activities [22, 93, 111, 121, 287]. These tasks encompass the fine-grained motor and tactile activity components.

Emergency response tasks focus on medical procedures (e.g., cardiopulmonary resuscitation and chest-tube decompression) performed on patients seeking emergency medical attention [88, 154, 157]. These tasks are composite in nature and typically involve the gross motor, fine-grained motor, and tactile activity components.

Fitness tasks include recreational and sports activities [52, 62, 128]. These tasks primarily involve the gross motor, followed by the fine-grained motor activity components.

2.2 Task Recognition Metrics Evaluation

Task recognition algorithms require metrics to assess the tasks performed by individuals; therefore, selecting appropriate sensing metrics is crucial. The metrics employed in a task recognition algorithm depend on the task domain and the activity components involved in the task set. Thirty two task recognition metrics were identified across the activity components. The metrics are evaluated using the following criteria: sensitivity, versatility, and suitability. Directly comparing task recognition metrics is challenging, as the metrics are employed in different task recognition algorithms across different task domains containing different task types. Thus, the metrics' classifications are provisional, although the classifications for widely used metrics (e.g., inertial metrics) are unlikely to change.

2.2.1 Evaluation Criteria

Sensitivity refers to a metric's ability to detect tasks reliably. A metric's sensitivity is classified as *High* if at least three citations indicate that the metric detects tasks with $\geq 80\%$ accuracy, while a metric is classified as *Medium* if the task detection accuracy is $\geq 70\%$, but $< 80\%$. *Low* metric sensitivity occurs if the metric detects tasks with $< 70\%$ accuracy. Metrics without sufficient citations to determine their sensitivity are classified as *Indeterminate*, and additional evidence is required to substantiate the metric's sensitivity.

Versatility refers to a metric's ability to detect tasks across different task domains. A metric's versatility is *High* if the metric is cited for discriminating tasks in at least two or more task domains. Similarly, if the metric was used for classifying tasks belonging to only one task domain, the versatility is *Low*.

Suitability evaluates the feasibility of using a metric for detecting tasks in various physical environments (i.e., structured vs. unstructured), which depends on the sensor technology employed to gather the metric. Ideally, sensors must be independent of the environment and must be unaffected by any form of disturbance, so that they can be used for gathering data across domains and environments. Disturbances can occur in two ways: i) internal, and ii) external. *Internal disturbances* are intrinsic to the user (e.g., noise caused due to sensor displacement, or excessive perspiration during the task). *External disturbances* are caused due to changes in the environment (e.g., background noise or lighting conditions). Two sub-criteria were developed in order to evaluate a sensor for a metric's suitability: i) *wearability*, and ii) *reliability*. A sensor is classified as *wearable* if it can be worn by the user (e.g., accelerometers and gyroscopes), while sensors classified as *unwearable* (i.e., environmentally embedded sensors) are mounted at fixed locations throughout the environment. Acoustic [141], ambient [121, 268], and other static sensors (e.g., cameras [114, 130] and Radio Frequency Identification (RFID) tags [118, 157]) are a few examples of environmentally embedded sensors. A sensor is classified as *reliable* if it is unaffected by both external and internal disturbances, while a sensor is classified as *unreliable* if it is susceptible to internal or external disturbances. These two subcriteria inform the suitability of a metric.

A metric's *suitability* can be classified as conforming, or non-conforming, where conforming is defined as complying with the criterion. A metric is classified as *conforming* if the metric can be gathered by a sensor that is both *wearable* and *reliable*, while a metric's suitability is classified as *non-conforming* if the metric cannot be gathered by a *wearable,*

reliable sensor. A metric conforming with the suitability criteria indicates that it can be used to detect tasks in both structured and unstructured environments. Conversely, a metric not conforming with the suitability criteria is typically limited to detecting tasks only in structured environments.

The task recognition metrics and the corresponding sensitivity, versatility, and suitability classifications are provided in Table 2.1. The *Activity Component* column in Table 2.1 indicates which components (i.e., cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile) are associated with the metric. For example, the inertial metrics are primarily used for recognizing gross and fine-grained motor tasks (e.g., grasping and object manipulation [127, 130, 185]), while electrooculography metric is widely employed for recognizing visual tasks (e.g., reading from a paper, or a computer [30, 99]). Certain metrics may not measure an activity component directly, but provide contextual information that can be leveraged to inform the performed task. For example, RFID, which measures a user's close proximity to a particular object or location, can indicate a human's interaction with the object [94] or involvement in an activity associated with the location [157], both representing contextual information that can aid task recognition.

2.2.2 Task Recognition Metrics

A classification of the task recognition metrics based on their sensing properties is presented. Further, each metric is evaluated based on three evaluation criteria in order to identify the most reliable, minimal set of metrics to recognize tasks for the intended HRT domain.

2.2.2.1 Inertial Metrics

Inertial metrics can be decomposed into two independent metrics: i) linear acceleration measures the three-dimensional acceleration of a body region via an accelerometer; and ii) orientation, or a body part's rotation and rotational rate in three-dimensions, as gathered using gyroscope and magnetometer. Inertial metrics can be gathered jointly using an Inertial Measurement Unit (IMU) sensor that integrates the accelerometer, gyroscope and magnetometer into a single sensor. Inertial metrics are primarily used for detecting physical tasks, which includes both gross and fine-grained motor tasks [45, 47, 146, 266].

Linear acceleration is the most widely employed inertial metric, and can be used as a

Table 2.1: Metrics evaluation overview by Sensitivity (Sens.), Versatility (Verst.), and Suitability (Suit.), where $\bigvee$, $\prod$, and $\bigwedge$, represent Low, Medium, and High, respectively. (.) indicates Indeterminate, while * indicate hypothesis predicted for the particular metric. Suitability is classified as conforming (C) or non-conforming (NC).

Category	Metrics	Sens.	Verst.	Suit.	Activity Component
Inertial	Acceleration	$\bigwedge$	$\bigwedge$	C	Gross [10, 43, 135, 151, 224]
					Fine-grained [142, 272, 282]
					Tactile [35, 109, 166]
	Orientation	$(\bigwedge^*)$	$\bigwedge$	C	Gross [62, 105, 185]
					Fine-grained [127, 128, 185]
Eye Gaze	Fixation	$\bigwedge$	$\bigwedge$	C	Visual [116, 140, 254]
	Saccades	$\prod$	$\bigwedge$	C	Visual [30, 99, 253]
	Scanpath	$\prod$	$\bigvee$	C	Visual [100, 174, 253]
	Lookahead fixation	$(\bigwedge^*)$	$(\bigwedge^*)$	C	Visual* [211, 246]
	Blink rate	$(\bigvee^*)$	$\bigwedge^*$	C	Visual [30, 97]
		$(\bigwedge^*)$	$\bigwedge^*$	C	Cognitive* [85, 173]
	Pupil dilation	$(\bigwedge^*)$	$(\bigwedge^*)$	NC	Visual* [217, 221, 267]
					Cognitive* [7, 85, 173]
Electro-physiological	EOG	$\prod$	$\bigwedge$	NC	Visual [30, 98, 100]
		$(\bigwedge^*)$	$\bigwedge$	NC	Cognitive [138, 168]
	sEMG	$\bigwedge$	$\bigwedge$	NC	Gross [202, 241, 261]
					Fine-grained [22, 59, 62, 111, 128]
		$(\bigwedge^*)$	$\bigwedge$	NC	Tactile [41, 42, 284]
	ECG potential	$\bigvee$	$\bigvee$	C	Gross [208, 210, 255]
	Heart rate	$\bigvee$	$\bigvee$	C	Gross [192, 205, 258]
	Heart rate variability	$(\bigwedge^*)$	$(\bigwedge^*)$	C	Cognitive* [85]
	EEG event-related potential	$\bigwedge$	$\bigvee$	NC	Cognitive [75, 234, 286]
	EEG Power spectral density	$\bigwedge$	$\bigvee$	NC	Cognitive [133, 188, 238]
Vision-based	Optical flow	$\bigwedge$	$\bigwedge$	NC	Gross [125, 131, 264]
		$\prod$	$\bigwedge$	NC	Fine-grained [137, 264]
	Human-body pose	$\prod$	$\bigwedge$	NC	Gross [104, 163, 193]
		$\bigwedge$	$\bigwedge$	NC	Fine-grained [104, 154, 163, 193]
	Object detection	$\bigvee$	$\bigwedge$	NC	Gross [58, 176, 189, 216]
					Fine-grained [58, 66, 216, 274]
Acoustic	Spectrogram	$\bigwedge$	$\bigwedge$	C	Auditory [82, 144, 159]
	MFCC	$\bigwedge$	$\bigwedge$	C	Auditory [184, 256, 279]
Speech	Transcript	$(\prod)^*$	$(\bigvee)^*$	C	Speech [78]
	Keywords	$(\prod)^*$	$(\bigvee)^*$	C	Speech [6]
	Speech rate	$(\bigwedge)^*$	$(\bigwedge)^*$	C	Speech [61, 85, 87]
	Voice intensity	$(\bigwedge)^*$	$(\bigwedge)^*$	C	Speech [61, 87]
	Voice pitch	$(\bigwedge)^*$	$(\bigwedge)^*$	C	Speech [61, 87]
Localization	Outdoor localization	$\bigvee$	$\bigwedge$	NC	Gross [162, 228, 293]
	Indoor localization	$\bigwedge$	$\bigwedge$	NC	Gross [57, 94, 196]
					Fine-grained [60, 156, 157]
Miscellaneous	Physiological	$\bigvee$	$\bigwedge$	C	Gross [147, 185, 206]
	Contact forces	$(\bigwedge^*)$	$(\bigwedge^*)$	C	Tactile [93, 111]
	Elec. Impedance Tomography	$(\bigvee^*)$	$\bigvee$	NC	Fine-grained [287]
	Electromagnetic noise	$(\bigwedge^*)$	$\bigvee$	NC	Fine-grained [143, 269]

standalone task recognition metric. Orientation is often used in combination with linear acceleration for task recognition, but can also be used independently. The type and number of tasks detected by the inertial metrics can be linked to the number and placement of the sensors on the body [15, 45]. For example, inertial metrics for detecting gross motor tasks involve placing the sensors at central or lower body locations (e.g., chest, waist, and thighs [70, 135, 172, 224]), while detecting fine-grained motor tasks involve placing the sensors at the forearms and wrists [88, 127, 185, 282], and tactile task recognition places the sensors [35, 109, 166] at the hand's dorsal side and fingers. Linear acceleration has high sensitivity, as it can detect tasks with $\geq$ 80% accuracy. Standalone orientation is indeterminate, but it is hypothesized to have a high sensitivity. IMUs, accelerometers, gyroscopes, and magnetometers that gather inertial metrics can be *wearable*. The inertial metrics, with appropriate drift removal methods [19, 64], depend only on the users' movements, making them *reliable*. The inertial metrics can detect tasks across multiple task domains. Thus, inertial metrics have high versatility and conform with suitability criteria.

2.2.2.2 Eye Gaze Metrics

Eye gaze metrics record the coordinates, g_x and g_y, of the point of gaze over time. Raw eye gaze data is often processed to yield more meaningful eye movement metrics that are representative of a user's visual behavior and can be leveraged for task recognition. Fixations, saccades, scanpath, blink rate, pupil dilation, and lookahead fixations are some of the most important eye movement metrics that can be extracted from eye gaze. Most eye gaze-based task recognition approaches use one or more of these eye movement metrics [92, 116, 134, 140, 174, 253, 254]. These metrics are commonly used for recognizing visual tasks, while some prior research has also detected cognitive tasks [100, 132].

Fixations are stationary eye states during which gaze is held upon a particular location [30], while the simultaneous movement of both eyes between two fixations is called a **saccade** [30]. Fixation has high sensitivity (e.g., [116, 140, 254]), while saccade has medium sensitivity (e.g., [30, 99, 253]). Both metrics have high versatility, and wearable eye trackers are reliable to disturbances; therefore, the two metrics conform with suitability.

A fixation-saccade-fixation sequence is called a **scanpath** [174]. Scanpath has medium sensitivity, as it can only detect tasks within 70-80% accuracy (e.g., [100, 174, 253]). Scanpath has a low versatility, because, it has been used only for detecting desktop or office-based visual activities (e.g., reading a document, web browsing, writing, and typing).

Scanpath conforms with the suitability criteria.

Blink rate is defined as the number of blinks (i.e., opening and closing eyelids) per unit time [30]. Blink rate is often used in conjunction with fixation and saccade as an additional metric to provide further context. The metric's sensitivity is classified as indeterminate due to lack of sufficient standalone citations, but it is hypothesized to have low sensitivity for visual task recognition and is used predominantly to detect desktop or office-based visual activities (e.g., [30, 97]). Additionally, prior research indicates that the metric highly correlates with the cognitive workload in supervisory task domains [85, 173]; therefore, blink rate is hypothesized to have high sensitivity toward cognitive task recognition in the supervisory task domain. The metric is also hypothesized to have high versatility, as it can be potentially used in multiple task domains. Finally, the metric conforms with suitability, as it can be measured using wearable eye trackers.

Pupil dilation is the measure of change in pupil diameter. The metric's sensitivity is indeterminate; however, based on the metric's ability to reliably detect cognitive workload [7, 85, 173], and its high correlation in various visual search tasks [217, 221, 267], it is hypothesized to have high sensitivity and versatility toward cognitive and visual task recognition. Pupil dilation requires precise measurements on the order of tenths of a millimeter. Lighting changes can significantly impact the metric's acquisition, so it does not conform with suitability.

Lookahead fixations are anticipatory eye movements, where humans fixate on objects that will be interacted with several seconds into the future [180]. The lookahead fixation metric analyzes a human's oculomotor behavior when performing tasks that typically require planning for multiple time steps into the future (e.g., driving, reaching to grasp an item; and manipulating objects). Lookahead fixation has never been used for task recognition, so its sensitivity is indeterminate. However, it is hypothesized to have high sensitivity and versatility to detect visual tasks when used in conjunction with fixation and saccade metrics, especially for tasks involving high hand-eye coordination [211, 246]. The metric conforms with suitability, as it can be measured by wearable eye trackers.

2.2.2.3 Electrophysiological Metrics

The electrical signals associated with the nervous system and other body parts (e.g., muscles and eyes, are called electrophysiological signals. These signals can be leveraged for task recognition, as they are highly correlated with activities humans conduct. The most

common electrophysiological metrics are electromyography, electrocardiography, electroencephalography, and electrooculography.

Electromyography measures the time-varying voltage signal produced by muscle tissues during contraction and relaxation. Surface-Electromyography (sEMG) is a non-invasive electromyography technique, wherein electrodes placed on the skin overlying a muscle measure the electrical activity. sEMG is the most commonly employed electromyography technique in task recognition systems [22, 59, 130]. sEMG electrodes are typically placed either on the forearm or upper limb, depending on the tasks to be detected. Forearm positioned sEMG commonly involves detecting fine-grained motor tasks (e.g., [62, 88, 111, 128]), while upper limb positioned sEMG involves detecting gross motor tasks (e.g., [241, 261]). The repeatability and reliability of the sEMG metric for detecting tasks depends on the number of electrodes (or channels) used to gather the metric. The majority of the research used 8-channel sEMG sensors (e.g., [59, 62, 88, 128]), while 16-channel (e.g., [22, 111]) and <8-channel sEMG sensors (e.g., [41, 42, 202, 261, 284]) have also been used. The sEMG metric can detect tasks across various domains (e.g., industrial [130], emergency medical response [88], and fitness [128]); thus, making it the most widely adopted electrophysiological metric for detecting gross motor and fine-grained motor tasks. The metric has high sensitivity, and high versatility for detecting gross and fine-grained motor tasks. The metric has been employed for detecting various finger and intricate hand motions (e.g., [41, 42, 284]); thus, it is hypothesized to have high sensitivity for detecting tactile tasks. Finally, the metric does not conform with suitability, as sweat accumulation underneath the electrodes may compromise the sEMG sensor's adherence to the skin, as well as signal fidelity [4].

The **Electrooculography (**EOG) metric records the electrical activity caused by eye movements and measures the voltage between the front and back of the human eye. The metric has medium sensitivity for classifying visual tasks (e.g., typing, web browsing, reading and watching videos [30, 98, 100]). The metric is also capable of detecting cognitive tasks (e.g., [50, 138]), but its sensitivity for the cognitive activity component is indeterminate, due to insufficient citations. The metric has high versatility, as it can be used in multiple task domains [50, 138]. Although wearable glasses can measure EOG, the metric does not conform with suitability, because EOG data is susceptible to noise introduced by facial muscle movements [138].

Electroencephalography (EEG) collects electrical neurophysiological signals from different parts of the brain. EEG measures two different metrics: i) Event-related potential

measures the voltage signal produced by the brain in response to a stimulus (e.g., [75, 234, 286]); and ii) Power spectral density measures the power present in the EEG signal spectrum (i.e., alpha (8-12 Hz), beta (13-30 Hz), theta (4-8 Hz), and delta (< 4 Hz)). The power spectral density metric reflects humans' cognitive and memory performance [124] that can be leveraged for cognitive task recognition [133, 188]. Both EEG metrics have high sensitivity (e.g., [75, 133, 234]). EEG signals may not be accurate when a participant is physically active, so EEG is best suited for detecting cognitive tasks in supervisory-based environments, as well as office or desktop-based environments, where the participants are more sedentary. Therefore, the EEG metrics have low versatility. EEG signals suffer from low signal-to-noise ratios [243, 286], and incorrect sensor placement can create inaccuracies; therefore, EEG metrics do not conform with suitability.

Electrocardiography (ECG) measures the time-varying voltage signal that corresponds to the electrical activity of the heart beat. Similar to sEMG, ambulatory ECG is a non-invasive wearable methodology to record ECG potential [255]. ECG potential's signal artifacts induced by body movements [208, 210] can be leveraged to detect gross motor tasks (e.g., standing, walking, and climbing [209]). ECG potentials are not sensitive enough to detect tasks on their own; therefore, they are often used in conjunction with inertial metrics to detect gross motor tasks [107, 119]. ECG potential has low versatility, as it has been used to detect only ambulatory tasks (e.g., [107, 119, 210]). ECG potential conforms with the suitability criteria, as it can be gathered by a wearable sensor immune to environmental noise.

The ECG signals can also be used to measure two other heart-related metrics: i) heart rate, and ii) heart rate variability. **Heart rate** refers to the number of heartbeats per minute, while **Heart Rate Variability (**HRV) measures the variation in the heart rate's beat-to-beat interval. Heart rate has low sensitivity for detecting gross motor tasks (e.g., [192, 205, 258]) and is often used for distinguishing the participants' intensity when performing physical activities [192, 258]. Heart rate has low versatility, as it can only detect ambulatory tasks. The heart rate metric conforms with suitability, if an individual's stress and fatigue levels remain constant. The HRV metric is seldom used for task recognition [205], but it is sensitive to large variations in cognitive workload [85]. Therefore, the metric is hypothesized to have high sensitivity for cognitive task recognition. The metric conforms with suitability, and is hypothesized to have high versatility.

2.2.2.4 Vision-Based Metrics

Vision-based metrics use videos and images containing human motions in order to infer the tasks performed. Vision-based metrics are acquired by static cameras installed at fixed locations in the environment, or using wearable cameras that are typically mounted on a human's shoulders, head, or chest (e.g., [58, 176, 177]) to capture egocentric perspectives [114]. Such metrics provide rich information necessary to classify humans' interactions with objects, which allows for a high-level abstraction in task recognition (e.g., distinguishing between drinking coffee and drinking tea [47]). Video enables object detection, localization, and motion tracking [51, 84, 226], which are relevant for task recognition; however, its use is still discouraged. Environmentally embedded cameras will not always be available in unstructured domains or environments (i.e., outdoors). Even wearable cameras have limitations given that the video is still highly susceptible to background noise from lighting, vibrations, and occlusion. Further, recording videos raises privacy concerns [146], and video processing can be computationally expensive. These limitations constrain the use of vision-based metrics for the intended HRT domain.

Several vision-based metrics exist. **Optical flow** measures the relative motion of pixels between two image sequences. This metric has high sensitivity for gross motor task recognition (e.g., [125, 131, 264]), and medium sensitivity for detecting fine-grained motor tasks (e.g., [137, 264]). Optical flow is versatile, as it can be used in multiple task domains. The metric is non-conforming for suitability, as it is susceptible to vibrations and camera distortions. The metric is reliable only when the sensor is environmentally embedded.

Human-body pose measures the skeletal joint positions of an individual [88], which is highly correlated with the performed task [27]. The human-body pose metric has high versatility. The metric has medium sensitivity for detecting gross motor tasks (e.g., [104, 193]), and high sensitivity for fine-grained motor tasks (e.g.,[154, 163]). The metric can be measured using multiple sensors (e.g., IMUs, cameras, and depth sensors [56]); thus, the suitability criteria depends on the associated sensor.

Object detection refers to identifying and locating instances of objects of interest in an image or video. Object detection may not be involved in task recognition directly, but it provides relevant context to inform the performed task. Detecting an object that a human interacts with vastly reduces the number of possible tasks the human performs; thus, aiding task detection. Object detection has low sensitivity and high versatility for detecting gross and fine-grained motor tasks (e.g., [189, 274]). Some research combined human-body pose

and object detection in that they simultaneously tracked hand poses and detected objects in the hand in first-person vision to infer the tasks (e.g., [58, 66, 216, 278]). Finally, the metric is non-conforming for suitability, as they are highly susceptible to change in lighting conditions and occlusion.

2.2.2.5 Acoustic Metrics

Tasks performed by humans are typically accompanied by characteristic sounds [181]. Acoustic metrics leverage these characteristic sounds in order to detect auditory tasks in the surrounding environment. Audio signals are usually analyzed in the frequency-domain rather than the time domain, because the raw audio signal (i.e., time-domain signal) is too noisy and erratic to be useful. The frequency-domain signal is obtained by applying a Fourier transformation to the time-domain audio signal. Auditory task recognition algorithms commonly employ two types of frequency-domain metrics: i) spectrogram; and ii) cepstral coefficients.

Human ears are more sensitive to changes in sound at lower frequencies. The Mel scale is a frequency measurement scale introduced to relate the perceived frequency of a sound to the actual frequency measured in Hertz (Hz) [36]. A frequency measured in Hz can be converted to the Mel scale by performing a non-linear logarithmic transformation. Both spectrogram and cepstral coefficients are computed on the Mel-frequency scale instead of Hz, because the Mel-frequency scale better resembles the resolution of the human auditory system [36].

A **spectrogram** is a three-dimensional acoustic metric with the first dimension representing time, the second dimension representing frequency, and the third dimension indicating the amplitude of a particular frequency at a particular time. The Mel spectrogram's amplitude is also log-transformed sometimes for numerical stability [82]. The spectrogram provides high sensitivity and versatility (e.g., [82, 144, 159]).

The Mel-frequency Cepstrum represents the short-term power spectrum of a sound, and is obtained by applying a discrete cosine transformation on a log power spectrum of a sound wave on the Mel scale. The **Mel-Frequency Cepstral Coefficients (**MFCCs), which represent the amplitudes of the resulting cepstrum, are an acoustic metric employed in many auditory task detection algorithms. The metric has high sensitivity and versatility (e.g., [184, 256, 279]). Both metrics conform with suitability, as long as the audio is captured via a wearable microphone.

Noise level is another acoustic metric and measures the loudness of a task environment. The metric is measured in decibels using a sound decibel meter. Noise level correlates to an increase in auditory workload [86], but it has not been used for task recognition; therefore, additional evidence is required to substantiate its evaluation criteria. The metric is hypothesized to detect auditory tasks when the events are fewer (≤ 3). The main limitations are that it cannot differentiate between multiple auditory tasks occurring simultaneously and may fail if the auditory tasks have a similar loudness profile. The metrics' sensitivity and versatility are hypothesized to be low and high, respectively. The suitability criterion is classified as non-conforming, as the device cannot be worn by a human teammate. A viable alternative will be to mount the device on a robot teammate.

2.2.2.6 Speech Metrics

Communication exchanges between human teammates can be translated into text, or a **Verbal transcript**, such that the message is captured as it was spoken. Transcripts can be generated manually [78], or using an automatic speech recognition tool (e.g., SPHINX [149], Kaldi [218], Wav2Letter++ [220]). The transcribed words are encoded into $n-$dimensional vectors (e.g., *GloVe* vector embeddings [213]) to be used as inputs for detecting speech-reliant tasks [78]. Representative **keywords** that are spoken more frequently can be used for detecting tasks [6]. Keywords are detected for every utterance automatically using word-spotting software [65, 262]. Identifying keywords for each task is non-trivial and requires considerable human effort. Both transcript and keywords metrics' sensitivity is classified as indeterminate due to insufficient citations, but is hypothesized to be medium [6, 78]. The metrics conform with suitability, provided the speech audio is obtained using a wearable microphone. The metrics are highly domain specific; therefore, their versatility is hypothesized to be low.

Several speech-related metrics (e.g., speech rate, pitch, and voice intensity) that do not rely on natural language processing have demonstrated effectiveness for estimating speech-workload [61, 85, 87]. **Speech rate** captures the articulation and pause rate of verbal communications and is measured by the number of syllables per unit time [61]. **Voice intensity** is the root-mean-square value of the mono-channel audio signal, while **Pitch** is the signal's dominant frequency over a time period [87]. These metrics have not been used for task recognition; therefore, additional evidence is required to substantiate their evaluation criteria. These metrics are hypothesized to detect speech-reliant tasks based on

their ability to estimate speech workload [61, 87]. The metrics may fail to detect the tasks when the speech component is short durationed (e.g., one word or one syllable replies), as the metrics do not offer the same amount of task-appropriate context when compared to *keywords* or *verbal transcripts*; however, the metrics are believed to provide evidence for tasks with multiple activity components. For example, a communication response to an air traffic radio message is predominantly a speech-reliant task, but may be accompanied by a *physical action* (e.g., changing the radio frequency). The task may not be detected solely by using speech-related metrics, but the metrics add validity by providing additional context to the *physical action*, which in this case is a fine-grained motor component. The metrics' sensitivity and versatility are hypothesized to be high. The suitability criterion is classified as conforming, assuming that the speech audio is obtained via wearable microphones.

2.2.2.7 Localization-Based Metrics

Localization-based metrics infer the task performed by analyzing either the absolute or relative position of items of interest, including humans. Localization-based metrics can be of two different types: i) **Outdoor Localization**, and ii) **Indoor Localization**. The outdoor localization metric often involves the human wearing a receiver, which relies on satellite navigation systems (e.g., Global Positioning System) to determine the absolute location (i.e., in latitude and longitude coordinates) of humans. The outdoor localization metric has low sensitivity, as the knowledge of location alone cannot determine the performed tasks accurately [162], but can support task recognition by providing context [225, 228]. Therefore, the metric is often paired with inertial metrics for task recognition [10, 146, 293]. The metric is highly versatile, as it has been used in multiple task domains (e.g., ADL and fitness). A wearable sensor can measure the metric, but can be unreliable, due to the inherent error (on the order of meters) present in the measurement. Events that affect the penetration of satellite signals (e.g., adverse weather conditions, dense tree canopy, and indoor environments) can also result in inaccurate localization [146]. Thus, the metric does not conform with suitability.

Indoor localization determines the relative position of items, including humans, with respect to a reference in indoor environments by acquiring the change in radio signals. RFID tags and wireless modems installed at stationary locations are the standard sensors. The indoor localization metric infers tasks by determining humans' location or identifying objects lying in close proximity [47, 57]. For example, reaching for a blender equipped

with an RFID tag indicates that the human is about to operate the blender. The indoor localization metric can detect gross motor and fine-grained motor tasks in multiple task domains with $\geq 80\%$ accuracy (e.g., [57, 60, 94, 156, 157, 196]); therefore, the metric has high sensitivity and versatility. The metric requires environmentally embedded sensors; therefore, it is non-conforming for suitability.

2.2.2.8 Miscellaneous Metrics

Physiological metrics provide precise information about an individual's state. Several physiological metrics exist: i) **Galvanic skin response**, which measures the conductivity of the skin, ii) **Respiration rate**, which represents the number of breaths taken per minute, iii) **Posture Magnitude**, which measures an individual's trunk flexion (leaning forward) and extension (leaning backward) angle in degrees, and iv) **Skin temperature**, are the most commonly used physiological metrics for task recognition. The physiological metrics have low sensitivity toward task recognition, since they react to task changes with a time delay. The physiological signals correlate with the intensity level of the activity, but they do not reflect the type of activity [206]; however, they may improve recognition accuracy when used as auxiliary metrics with inertial data [147, 185]. The metrics have high versatility, as they have been used in multiple task domains, although they are used predominantly to detect gross motor tasks. The physiological metrics conform with suitability, as they can be measured by wearable, reliable sensors.

Humans are particularly receptive to tactile cues on their hands, limbs, feet and torso, which allow complex tasks to be carried out [169]. Therefore, understanding tactile interactions are fundamental to contextual task recognition. Prior tactile task recognition research employed sensors for detecting only hand-based tactile interaction (e.g., [42, 93, 109, 111]). Luo et al. [169] developed a wearable full-body tactile textile that can capture tactile cues on any body part by quantifying the contact forces.

Contact forces are a tactile metric that senses grasp, touch, and other forces that arise when humans interact with the environment physically [111, 201]. The contact forces metric is hypothesized to be highly sensitive for detecting tactile-oriented tasks that involve object interaction and manipulation [93, 111]. The metric is hypothesized to be highly versatile, as it can be used in multiple different task domains, as long the tasks involve object interaction. The metric conforms with suitability, as it can be measured by a reliable wearable sensor; however, the sensor employed may require wearing a pair of gloves or a

full-body suit, which can hinder task performance.

Most electrical and electromechanical objects (e.g., computing devices, power tools, and automobiles) emit a signature **Electromagnetic noise** [269]. When a human makes physical contact with such objects, the emitted electromagnetic signals that propagate through the user's body can be used to identify the objects, which in turn can inform the activity [143]. However, the metric is limited to only objects that generate electromagnetic noise. Reproducibility is also an issue, since the sensors are not available commercially. Tomography is a non-invasive method of analyzing objects' inner structure and composition with radiation (e.g., Computerized Tomography scans and Magnetic Resonance Imaging) [23]. For example, **Electrical Impedance Tomography** [90] leverages muscle tomography (e.g., change in muscles' cross-sectional shape and impedance distribution when flexed) to recognize hand gestures [287]. Both Electrical Impedance Tomography and Electromagnetic noise metrics do not have adequate citations to inform sensitivity, and are classified as indeterminate. The metrics have only been cited for detecting ADL tasks, so they have low versatility. The associated wearable sensors were developed and used in laboratory settings; thus, the reliability is not completely understood. Overall, these metrics are non-conforming for suitability.

2.2.3 Discussion

The objective of any task recognition algorithm is to detect tasks encompassing multiple activity components via a reliable, but minimal set of metrics. Ideally, a metric must detect tasks with high accuracy and repeatability. The metrics and associated sensors best suited for each activity component that can be employed across multiple task domains and environments were reviewed. HRTs must operate in dynamic, unstructured environments, emphasizing the need for gathering relevant metrics with unobtrusive wearable sensors that do not impede humans' movement, and are resilient to environmental characteristics (e.g., humidity, heat). The intended domain's tasks involve all seven activity components; thus, identifying metrics capable of detecting tasks across activity components given the environment uncertainty is crucial.

The *gross motor* tasks can be detected accurately using inertial metrics measured at the waist, shoulder, thigh and ankle locations, while sEMG metrics in conjunction with inertial metrics can be used to recognize *fine-grained motor* activities. Additional context for physical tasks can be provided by the human-body pose metric estimated using

IMUs. Hand-based *tactile* tasks can be detected using contact forces; however, instrumenting fingertips with gloves may render the metric infeasible for certain task domains and environments. A suitable alternative is to measure acceleration at the dorsal side of the hand using accelerometer sensors. Therefore, the suitable sensor suite to recognize the gross, fine-grained motor, and tactile tasks correspond to a wearable inertial-based motion tracking system (e.g., Xsens MTw Awinda [207]) that measures the inertial metrics at various body parts, and can also uses the measurements to compute the human-body pose metric. Pairing the motion tracking system with a wearable forearm sEMG measurement sensor (e.g., Myo Armband [239]) can enable a task recognition algorithm to differentiate highly articulated fine-grained motions.

Visual tasks are best detected using fixation, saccade, and scanpath metrics captured via eye tracking. The lookahead fixation metric may be used to provide additional context if the tasks involve hand-eye coordination. The *cognitive* tasks are best detected using EEG metrics (i.e., power spectral density or event-related); however, EEG signals are erratic and vulnerable to physical movements. Pupil dilation, blink latency, and blink rate metrics are hypothesized to be suitable alternatives for detecting cognitive tasks. Therefore, a wearable eye-tracker (e.g., Pupil Core eye-tracker) capable of measuring the aforementioned metrics can be employed to detect both visual and cognitive tasks, although further research is required to confirm the efficacy of pupil dilation and blink metrics toward detecting cognitive activities. Physiological metrics (e.g., HRV with at least 30 seconds of data) may also be used as substitute metrics to detect cognitive tasks.

None of the reviewed metrics detect *speech* tasks reliably due to the lack of sufficient research [102]. Among the reviewed metrics, keywords and transcripts are the only metrics that provide task appropriate context necessary for detecting speech-reliant tasks [6, 102], but at the cost of being highly domain-specific and requiring natural language processing (e.g, word-spotting and speech recognition). The manual effort required to extract sensitive keywords for each task along with the added privacy concerns discourage the use of these metrics. The other evaluated speech-related metrics (e.g., speech rate, voice intensity, pitch, and utterance length) are untested for task detection, but can potentially contribute to task identification based on the speech patterns associated with the tasks. Further, these metrics are not domain-specific and offer better anonymity by deemphasizing natural language processing. Finally, spectrogram and MFCCs measured from a wearable microphone can detect *auditory* events reliably.

2.3 Task Recognition Algorithms Evaluation

Over one hundred task recognition algorithms across different activity components and task domains were identified and reviewed. The algorithms are evaluated using the following criteria: sensitivity, suitability, generalizability, composite factor, concurrency, and anomaly awareness. The evaluation criteria and the corresponding requirements were chosen in order to assess an algorithm's viability for detecting tasks in a human-robot teaming domain. Directly comparing task recognition algorithms is challenging, as there exists no established criteria for comparing and evaluating them. The classifications compare the algorithms by activity components with similar tasks. An algorithm may require additional evidence in order to assess a particular criterion, if there is insufficient information to classify the algorithm or if the algorithm was not developed in a practical setting.

2.3.1 Evaluation Criteria

Sensitivity refers to an algorithm's ability to detect tasks reliably. An algorithm's sensitivity is classified as *High* if the algorithm detects tasks with $\geq 80\%$ accuracy, while *Medium* if the algorithm's accuracy is $\geq 70\%$, but $< 80\%$, and *Low* if the accuracy is $< 70\%$. The accuracy thresholds were chosen by fitting a skewed Gaussian curve on the reviewed task recognition algorithms' accuracies.

An algorithm's *suitability* evaluates its feasibility for detecting tasks in various physical environments (i.e., structured vs. unstructured). An algorithm's suitability can be classified as conforming or non-conforming. An algorithm conforms if it can detect tasks independent of the environment; thus, the suitability criterion is dependent on the incorporated task recognition metrics. An algorithm is conforming if it incorporates *wearable*, *reliable* metrics, and is non-conforming otherwise.

Generalizability represents an algorithm's ability to identify tasks across individuals. The generalizability criterion depends on the achieved accuracy, given the algorithm's validation method. An algorithm conforms if it achieves $\geq 80\%$ accuracy with *leave-one-subject-out* cross-validation or *in-the-wild* validation. The *leave-one-subject-out* cross-validation approach reports the average accuracy obtained by training the algorithm repeatedly on all but one participant's data and validating using the left-out participant's data. The *in-the-wild* validation approach reports the average accuracy obtained by validating on a new set of participants typically recruited after the algorithm's development,

in order to assess the algorithm's ecological validity and real-world performance.

The *composite factor* criterion determines whether an algorithm can detect tasks composed of multiple atomic tasks. If a detected task incorporates two or more atomic activities, then the algorithm conforms with the composite task criterion. Typically, long-duration tasks that incorporate multiple action sequences per task are composite in nature.

Many HRT scenarios require humans to perform two or more tasks concurrently. At each time step, tasks involving direct human interaction can be identified as foreground tasks, while all other active tasks can be classified as background tasks. *Concurrency* determines if the algorithm can detect tasks executed simultaneously. Concurrency has multiple forms: (i) a task may be initiated prior to completing a task, such that a portion of the task overlaps with the prior task (i.e, *interleaved tasks*), and (ii) multiple tasks performed at the same time (i.e., *simultaneous tasks*) [11, 165]. An algorithm conforms if it can detect at least one form of concurrency.

Anomaly Awareness determines an algorithm's ability to detect an out-of-class task instance, which arises when an algorithm encounters sensor data that does not correspond to any of the algorithm's learned tasks. An algorithm conforms with anomaly awareness if it can detect out-of-class instances.

Most task recognition algorithms can only detect a predefined set of atomic tasks and are unable to detect concurrent tasks or out-of-class instances [47, 146]. Thus, unless identified otherwise, the reviewed algorithms do not conform with composite factor, concurrency, and anomaly awareness.

2.3.2 Overview of Task Recognition Algorithm Categories

Task recognition algorithms typically incorporate supervised machine learning to identify the tasks from the sensor data. These algorithms can be grouped into several categories based on feature extraction, ability to handle uncertainty, and heuristics. Three common data-driven task recognition algorithm categories exist in the literature.

Classical machine learning relies on features extracted from raw sensor data to learn a prediction model. Classical approaches are suitable when there is sufficient domain knowledge to extract meaningful features, and the training dataset is small.

Deep learning avoids designing handcrafted features, learns the features automatically [73], and is generally suitable when a large amount of data is available for training the model. Deep learning approaches leverage data to extract high-level features, while simul-

taneously training a model to predict the tasks.

Probabilistic graphical models utilize probabilistic network structures (e.g., Bayesian Networks [54], Hidden Markov Models [44], Conditional Random Fields [265]) to model uncertainties and the tasks' temporal relationships, while also identifying concurrent, composite tasks. The data-driven models' primary limitations are that they i) cannot be interpreted easily, and ii) may require a large amount of training data to be robust enough to handle individual differences across humans and generalize across multiple domains.

Knowledge-driven task recognition models exploit heuristics and domain knowledge to recognize the tasks using reasoning-based approaches (e.g., ontology and first-order logic [231, 257, 260]). Knowledge-driven models are logically elegant and easier to interpret, but do not have enough expressive power to model uncertainties. Additionally, creating logical rules to model temporal relations becomes impractical when there are a large number of tasks with intricate relationships [39, 161].

2.3.3 Cognitive Tasks

Cognition describes mental processes, including reasoning, awareness, perception, knowledge, intuition, and judgment [138], as such, most tasks require some cognitive capability. For instance, although tasks, such as reading, writing, watching videos predominantly involve visual, fine-grained motor, or tactile components, they also entail a cognitive component. Therefore, it is impractical to disregard the cognitive task elements, but classifying all such tasks as cognitive is also infeasible. Thus, only those algorithms that explicitly mention identifying the tasks' cognitive aspect are reviewed. The evaluation of the reviewed cognitive task recognition algorithms is presented in Table 2.2.

2.3.3.1 Classical Machine Learning

EEG potentials, obtained by placing non-invasive electrodes on humans' scalp, are the primary electrophysiological metrics used to detect cognitive tasks. Features (e.g., amplitude and power spectral density) extracted from the EEG frequency bands (i.e., alpha (8-12 Hertz), beta (13-30 Hertz), theta (4-8 Hertz), and delta ($<$ 4 Hertz)) can be used to train classical machine learning algorithms [75, 133, 188]. Reading is the most widely detected cognitive task using EEG sensing. A k-Nearest Neighbors classifier distinguished between reading and non-reading tasks (e.g., drawing, watching a video and listening to music), as

Table 2.2: Cognitive task recognition algorithms' evaluation overview.

Category / Algorithm	Paper	Sens.	Suit.	Genr.	Comp.	Conc.	Anom.
Classical Machine Learning							
Decision trees	[97]	$\bigwedge$	C	NC	NC	NC	NC
Ensemble	[75]	$\bigwedge$	NC	NC	NC	NC	NC
k-Nearest Neighbors	[133]	$\prod$	NC	NC	NC	NC	NC
SVM	[138]	$\bigwedge$	NC	NC	NC	NC	NC
	[50]	$\bigwedge$	NC	NC	NC	NC	NC
Deep Learning							
CNN	[286]	$\bigwedge$	NC	NC	NC	NC	NC
	[238]	$\bigwedge$	NC	C	NC	NC	NC
CNN + LSTM	[234]	$\bigwedge$	NC	NC	NC	NC	NC

well as distinguishing reading different kinds of document, using a wearable EEG sensor [133].

EOG is the other electrophysiological metric employed for detecting cognitive tasks [50, 138]. The efficacy of different EOG features in detecting cognitive tasks was investigated [50]. Three features (i.e., adaptive autoregressive parameters, wavelet coefficients and Hjorth parameters) were extracted from a laboratory-developed two-channel EOG signal acquisition device. These features were used independently and in combinations to train a Support Vector Machine (SVM) classifier to detect eight cognitive tasks (e.g., reading, writing, copying a text, web browsing, watching a video, playing an online game, and word search).

Several other algorithms combine data from multiple sensing modalities to improve cognitive task recognition accuracy [75, 97, 138]. For example, combining blink rate and head motion by fusing the eye gaze data with acceleration data improved a decision tree classifier's accuracy in detecting four cognitive tasks (e.g., reading, solving a math problem, watching a video and talking) [97]. The *Codebook* algorithm recognized six cognitive tasks (e.g., reading a printed page, watching a video, engaging conversation, writing handwritten notes and sorting numbers) by clustering the subsequences sampled from a data sequence based on similarity [138]. The resulting cluster centers act as the set of codewords (i.e., codebook). A SVM classifier was trained to classify the histogram reflecting the codeword

frequency to predict the tasks.

Classical machine learning algorithms typically have high *sensitivity*; however, the incorporated metrics (i.e., EOG and EEG) are unreliable and cannot accommodate individual differences. Therefore, the algorithms' *suitability* and *generalizability* are non-conforming. The *composite factor* and *concurrency* are non-conforming as well.

2.3.3.2 Deep Learning

Recent deep learning advances facilitate detecting cognitive tasks using EEG potentials acquired from off-the-shelf, wireless, wearable EEG devices. Most EEG wearable devices (e.g., [1–3]) record prefrontal EEG signals, which are correlated to a human's intellectual, emotional and cognitive states [234]. A deep EEG network detected three cognitive tasks (e.g., reading, speaking, and watching a video) using data collected from a wearable EEG sensor's [2] two prefrontal EEG channels [234]. The hybrid deep learning algorithm incorporated a Convolutional Neural Network (CNN) to populate the feature maps from raw EEG potentials, followed by a Long Short-Term Memory (LSTM) network for modeling the temporal state of the EEG feature maps. Most existing EEG-based algorithms focus on application-specific classification algorithms, which may not translate to other domains. A transferable EEG-based cognitive task recognition algorithm that can adaptively support varying EEG channels as input and operate on a wide range of cognitive applications was developed [286]. The algorithm combined deep reinforcement learning with an attention mechanism to extract robust and distinct deep features.

Detecting cognitive tasks with fewer EEG sensors in an unconstrained, natural environment is a challenging task due to low signal-to-noise ratio, lack of baseline availability, change of baseline due to domain environment and individual differences, as well as uncontrolled mixing of various tasks [238]. A deep learning algorithm [238] revealed that the backward sensor selection [26] technique can reduce the sensor suite significantly (i.e., from nine probes to three) without compromising accuracy. Two deep neural networks, a deep belief network and a CNN, were trained using the EEG power spectral density to distinguish between listening and watching tasks.

Similar to the classical machine learning approaches, the deep learning algorithms also tend to have high *sensitivity*, but incorporate EEG metrics that suffer from low-signal-to-noise and individual differences; therefore, the algorithms' *suitability* and *generalizability* are non-conforming. Additionally, none of reviewed algorithms conform with the *concur-*

rency and *composite factor*.

2.3.3.3 Discussion

Generally, cognitive tasks can be classified with $> 80\%$ accuracy; therefore, the algorithms' typically have high *sensitivity*. Excluding Ishimaru et al.'s [97] decision tree classifier, none of the other algorithms conform with suitability, as the metrics employed were EEG or EOG. Thus, none of the discussed algorithms are appropriate for detecting cognitive tasks for the intended HRT domain. Given (a) that cognitive and visual tasks are closely associated, and (b) the efficacy of multimodality sensing [97], it is hypothesized that a classical machine learning algorithm that incorporates eye gaze metrics (e.g., pupil dilation, blink latency, blink rate) and cognitive workload sensitive physiological metrics (e.g., HRV) will be viable for detecting cognitive tasks.

2.3.4 Speech Tasks

Verbal communication plays a key role in task performance (e.g., assigning tasks, sharing or confirming important information, and reporting task completion), especially in a dynamic environment [290]. Speech-reliant task recognition in a highly dynamic environment (e.g., trauma resuscitation [24]) encounters several challenges (e.g., inconsistent verbal reports between tasks) the potentially succinct and non-grammatical nature of verbal communication, overlapping multi-person speech, and interleaved verbal exchanges due to multi-tasking [103]. Only two speech task recognition algorithms were identified, their classifications are cited in Table 2.3.

Table 2.3: Speech task recognition algorithms' evaluation overview.

Category / **Algorithm**	**Paper**	**Sens.**	**Suit.**	**Genr.**	**Comp.**	**Conc.**	**Anom.**
Deep Learning							
Attention	[78]	$\prod$	NC	NC	NC	NC	NC
CNN	[6]	$\bigvee$	NC	NC	NC	NC	NC

2.3.4.1 Deep Learning

Little research exists on speech-reliant task recognition [102], with the existing algorithms being based on deep learning (e.g., [6, 78]). A text-based task recognition algorithm employed a verbal transcript derived from a medical team's communication as input to predict ten trauma resuscitation tasks [78]. The algorithm used both speech and ambient sounds for task prediction. A multimodal attention network was applied to process the transcribed spoken language and the ambient sounds in order to predict the tasks. The main limitation was reliance on manually generated transcripts, which is infeasible for a contemporaneous task recognition system [5]. Automatic transcript generation requires a computationally expensive speech recognition tool. Additionally, poor audio quality caused by distant talking, ambient noise, succinct and non-grammatical speaking can increase an automatic speech recognition tool's error rate; thus, the algorithm's performance in real-world scenarios is expected to be lower than the cited result [78].

An alternative speech-reliant task recognition algorithm depended only on one keyword to detect trauma tasks [6]. The speech-reliant algorithm used one representative keyword per utterance as input to a deep neural network, in addition to the ambient sounds. This keyword was determined by calculating the most frequent words list for each task based on the premise that frequently occurring words for particular tasks can serve as features for the neural network's task prediction. Word-spotting tools (e.g., [65, 262]) can extract keywords efficiently, reducing the reliance on traditional speech recognition. The deep learning architecture consisted of an audio network, a keyword network, and a fusion network. The audio network adapted a modified *VGGish* deep network [249] to extract features from the audio spectrogram, while the keyword network extracted important verbal features from the keyword list. The fusion network concatenated the output of both networks to predict the speech-reliant tasks.

The use of speech to recognize tasks is an under-developed area. The reviewed algorithms had low to medium *sensitivity* (e.g., [6, 78]). The algorithms' *suitability* criterion was classified as non-conforming, because the incorporated metrics are domain and task specific. Both algorithms' are non-conforming with *generalizability*, *composite factor*, *concurrency* and *anomaly awareness*.

2.3.4.2 Discussion

Verbal communication in a highly dynamic setting (e.g., trauma medical team [290]) occurs at a high level (e.g., discussing task plans and intentions) and at a low level (e.g., coordinating, executing and reporting task completion) [102]. Identifying speech patterns and keywords during verbal communication can detect the tasks, as well as track their progress (i.e., preparing-performing-reporting). Incorporating verbal exchanges (e.g., transcripts [78] or keywords [6]) in tandem with the audio stream increased the accuracy by at least 15% in both algorithms, indicating that speech patterns and keywords can serve as differentiators for speech-reliant tasks. The algorithms' main limitations are that the metrics are highly domain-specific and require natural language processing, along with substantial manual effort to identify task specific sensitive keywords (see Section 2.2). Therefore, the algorithms cannot be readily transferred across domains. A suitable alternative may involve modifying the algorithm to use the audio stream in conjunction with speech workload metrics (e.g., speech rate, voice intensity, and voice pitch) instead of verbal exchanges.

2.3.5 Auditory Tasks

Auditory task recognition involves identifying characteristic ambient sounds in order to detect tasks in an environment [144, 181]. Auditory task recognition algorithms typically employ microphone sensors to detect sound events. Individual classifications for each auditory task detection algorithm by its category are provided in Table 2.4.

2.3.5.1 Classical Machine Learning

Auditory task detection algorithms that incorporate MFCCs typically use classical machine learning (e.g., [256], [279]), while deep learning algorithms incorporate spectrograms (e.g., [82, 91, 144, 159]). A Random Forest (RF) based voting algorithm, *Non-Markovian Ensemble Voting*, used MFCCs to recognize characteristic sounds produced by twenty-two ADL tasks [256]. The predictions were refined over time by collecting consensus via voting from past and future predictions.

A wearable acoustic sensor, *BodyScope*, worn around the neck classified several ADL tasks [279]. The *BodyScope* sensor contained a microphone surrounded by a stethoscope chest piece for sound amplifications in order to exploit the sounds that occurred at a

Table 2.4: Auditory task recognition algorithms' evaluation overview.

Category / Algorithm	Paper	Sens.	Suit.	Genr.	Comp.	Conc.	Anom.
Classical Machine Learning							
RF	[256]	$\bigwedge$	NC	C	NC	NC	NC
SVM	[279]	$\prod$	NC	NC	NC	NC	NC
Deep Learning							
CNN	[144]	$\bigwedge$	C	C	NC	NC	NC
	[91]	$\bigwedge$	RE	RE	NC	NC	NC
	[159]	$\bigvee$	NC	NC	NC	NC	NC
	[82]	$\prod$	C	NC	NC	C	NC
	[233]	$\prod$	RE	NC	NC	NC	NC

human's mouth and throat regions to recognize the tasks. For instance, when a person speaks to someone, they generate vocal sounds, while eating and drinking produce chewing, sipping, and swallowing sounds. Several time- and frequency-domain features (e.g., zero-crossing rate and MFCCs) were used to train an SVM classifier.

Both algorithms achieved $> 70\%$ accuracy during an in-the-wild study; therefore, the algorithms' *sensitivity* is medium to high. Although the metrics used were reliable, the ensemble voting algorithm incorporated an environmentally embedded microphone, while the *BodyScope* sensor suffers from non-reproducibility; therefore, the algorithms' *suitability* is classified as non-conforming. Finally, the algorithms' *composite factor* and *concurrency* are classified as non-conforming.

2.3.5.2 Deep Learning

Deep learning algorithms can leverage the time, frequency, and amplitude information in an audio signal's spectrogram to extract the spatio-temporal features. Most auditory task detection deep learning algorithms (e.g., [82, 144, 159]) leverage transfer learning. These algorithms fine-tune the existing *VGGish model* [91] (i.e., pre-trained on the *YouTube Audio Set* [67]) with additional layers to detect the target auditory tasks. The *VGGish model* used a log Mel spectrogram as input to output a 128-dimensional neural network feature vector for every second of an audio sample. A transfer learning framework detected fifteen

ADLs (e.g., talking, watching television, brushing, shaving, and listening to music) from the audio recorded using an off-the-shelf smartphone [159]. A five-layer CNN was added to the *VGGish*'s feature vector to predict the ADL tasks.

Polyphonic event detection algorithms recognize multiple auditory tasks occurring simultaneously [38], typically via deep neural networks. A *VGGish*-based algorithm stacked multiple binary classifiers to the feature vector [82], while others have developed various recurrent and hybrid deep neural networks to detect polyphonic sound events [31, 32, 204].

Audio augmentation can be exploited by deep learning to improve the recognition rate. Augmenting the original audio with a set of deformations (e.g., time stretching, pitch shifting, dynamic range compression, and background noise mixing) improved a CNN's classification accuracy significantly on a range of environmental sound classification tasks [233]. *Ubicoustics*, a real-time, auditory task recognition algorithm was trained by incorporating various augmentation techniques to simulate the sounds that resemble real-world audio samples [144].

Deep learning algorithms tend to outperform classical machine learning approaches in terms of accuracy. Thus, deep learning algorithms, especially the *VGGish*-based models [82, 144, 159], are the most suited for detecting auditory tasks, primarily due to their feature extraction capability and the availability of abundant audio datasets [67, 181]. Most algorithms are validated by splitting all the available data randomly into training and validation datasets; therefore, the algorithms' *generalizability* criteria either require additional evidence, or are non-conforming. All evaluated algorithms are non-conforming for the *composite factor* and *anomaly awareness* criteria.

2.3.5.3 Discussion

Most auditory task detection algorithms typically have medium to high *sensitivity* (e.g., [82, 91, 144, 256, 279]). The algorithms' *suitability* criterion depends on whether the microphone is worn or embedded in the environment. The algorithms' *generalizability* criteria either require additional evidence or are non-conforming. The polyphonic detection algorithms conform with concurrency; therefore, a deep polyphonic detection algorithm (e.g., [82]) is recommended for the intended HRT domain, as it is more likely to contain multiple, simultaneous sound sources [38]. None of the reviewed algorithms conform with *composite factor* and *anomaly awareness*.

2.3.6 Visual Tasks

Eye movement is closely associated with humans' goals, tasks, and intentions, as almost all tasks performed by humans involve visual observation. This association makes oculography a rich source of information for task recognition. Fixation, saccades, blink rate, and scanpaths are the most commonly used metrics for detecting visual tasks [30, 174, 253], followed by EOG potentials [98, 99, 168]. Visual tasks typically occur in *office or desktop-based* environments, where the participants are sedentary. The classifications of the reviewed visual task recognition algorithms are presented by algorithm category in Table 2.5.

Table 2.5: Visual task recognition algorithms' evaluation overview.

Category / **Algorithm**	**Paper**	**Sens.**	**Suit.**	**Genr.**	**Comp.**	**Conc.**	**Anom.**
Classical Machine Learning							
Auto-context model	[174]	$\bigwedge$	C	RE	NC	NC	NC
Decision trees	[140]	$\bigwedge$	C	C	NC	NC	NC
	[97]	$\bigwedge$	C	NC	NC	NC	NC
k- Nearest Neighbors	[98]	$\prod$	NC	NC	NC	NC	NC
RF	[253]	$\prod$	C	NC	NC	NC	NC
SVM	[116]	$\bigwedge$	C	NC	NC	NC	NC
	[30]	$\prod$	C	NC	NC	NC	NC
	[168]	$\bigwedge$	NC	C	NC	NC	NC
Deep Learning							
CNN	[100]	$\bigvee$	NC	NC	NC	NC	NC
CNN + LSTM	[99]	$\bigvee$	NC	NC	NC	NC	NC
Graph CNN	[139]	$\bigvee$	C	NC	NC	NC	NC
Encoder-Decoder	[182]	$\bigwedge$	C	C	NC	NC	NC

2.3.6.1 Classical Machine Learning

Classical machine learning using eye gaze metrics (e.g., saccades, fixation, and blink rate) for visual task recognition was pioneered by Bulling et al. [30]. Statistical features (e.g., mean, max, variance) extracted from the gaze metrics, as well as the character-based representation to encode eye movement patterns were used to train a SVM classifier to

detect five office-based tasks. The algorithm's primary limitation is that the classification is provided at each time instance t independently and does not integrate long-range contextual information continuously [174]. A temporal contextual learning algorithm, the *Auto-context* model, overcame this limitation by including the past and future decision values from the discriminative classifiers (e.g., SVM and k-Nearest Neighbors) recursively until convergence [174].

Low-level eye movement metrics (e.g., saccades and fixations) are versatile and easy to compute, but are vulnerable to overfitting, whereas high-level metrics (e.g., Area-of-Focus) may offer better abstraction, but require domain and environment knowledge [253]. These limitations can be mitigated by exploiting low-level metrics to yield *mid-level* metrics that provide additional context. The mid-level metrics were built on intuitions about expected task relevant eye movements. Two different mid-level metrics were identified: *shape-based pattern* and *distance-based pattern* [253]. The shape-based pattern metrics were based on encoding different combinations of saccade and scanpath, while distance-based pattern metrics were generated using consecutive fixations. The low- and mid-level metrics were combined to train a RF classifier to detect eight office-based tasks, including five desktop-based tasks and three software engineering tasks.

Various algorithms were developed focused solely on detecting reading tasks using classical machine learning (e.g., [116, 140]). The complexity of the reading task varied across algorithms. Reading detection can be as rudimentary as classifying active reading or not [140], or as complex as distinguishing between reading thoroughly vs. skimming text [116].

Based on feature mining, existing reading detection algorithms can be categorized into two methods: i) Global methods that mine eye movement metrics over an extended period ($> 30s$) to build a reading detector (e.g., [30, 99, 174, 253]), and ii) Local methods that extract the metrics within a narrow temporal window ($< 3s$) (e.g., [25, 126]). Global methods result in better accuracy, but do not detect reading in real-time, due to longer window sizes, while local methods allow for (near) real-time reading detection, but have low accuracy [116].

Classical machine learning algorithms (e.g., [30, 98, 140, 174, 253]) have medium to high sensitivity. Most algorithms conform with the *suitability* criterion, while rarely conforming with the *generalizability* criterion. All algorithms are non-conforming for the *concurrency* and *composite* factors, making them unsuitable for the intended HRT domain.

2.3.6.2 Deep Learning

Recent deep learning algorithms leverage CNNs to detect visual tasks directly using raw 2D gaze data obtained via wearable eye trackers. *GazeGraph* [139] algorithm converted 2D eye gaze sequences into a spatial-temporal graph representation that preserved important eye movement details, but rejected large irrelevant variations. A three-layered CNN trained on this representation detected various desktop and document reading tasks. An encoder-decoder based CNN detected seven mixed physical and visual tasks by combining 2D gaze data with head inertial metrics [182].

Several other algorithms apply deep learning techniques using EOG potentials to detect reading task [99, 100]. Two deep networks, a CNN and a LSTM, were developed to recognize reading in a natural setting (i.e., outside of the laboratory). Three metrics (i.e., blink rate, 2-channel EOG signals, and acceleration) from wearable EOG glasses were used to train the deep learning models.

Obtaining datasets at a large scale is difficult due to high annotation costs and human effort, while lack of labeled data inhibits deep learning methods' effectiveness. A sample efficient, *self-supervised CNN* detected reading task [100] using less labeled data. The self-supervised CNN employed a "pretext" task to bootstrap the network before training it for the actual target task. Three reading tasks (i.e., reading English documents, reading Japanese documents, both horizontally and vertically), as well as a no reading class, were detected by the self-supervised network. The pretext task recognized the transformation (i.e., rotational, translational, noise addition) applied to the input signal. The pretext pre-training phase initialized the network with good weights, which were fine-tuned by training the network on the target task (i.e., reading detection task) dataset.

The deep learning algorithms (e.g., [99, 100, 139]) typically tend to have low sensitivity. Further, the EOG deep learning algorithms do not conform with *suitability*, as the employed metrics are unreliable due to susceptibility to noise introduced by facial muscle movements [138]. These limitations discourage the use of deep learning for visual task recognition.

2.3.6.3 Discussion

None of the existing algorithms detected visual tasks within the targeted HRT context. The two classical machine learning algorithms: i) *Auto-context* model [174] and ii) Srivastava et al.'s [253] algorithm appear to be more appropriate for detecting visual tasks. Both

algorithms had $> 70\%$ accuracies across a range of visual tasks and employed eye gaze metrics; thus, conforming with *suitability* and partially with *sensitivity*. The algorithms' *generalizability* criterion requires additional evidence, as the former's validation scheme is unclear, while the latter does not have sufficient accuracy. None of the reviewed algorithms conform with the *concurrency* and *composite factor* criteria. The encoder-decoder algorithm [182] is also a viable alternative, as it achieved $> 80\%$ with leave-one-subject-out cross-validation using eye gaze metrics.

2.3.7 Gross Motor Tasks

Gross motor tasks occur across multiple task categories (e.g., *ADL, fitness*, and, *industrial*). A high-level overview of the reviewed algorithms with regard to the evaluation criteria is presented by algorithm category in Table 2.6.

2.3.7.1 Classical Machine Learning

Most gross motor task recognition algorithms incorporate classical machine learning using inertial metrics, often measured at central body locations [47], (e.g., chest [28, 55, 147, 155], waist [10, 14, 33, 135, 145, 224, 273, 283], and thighs [28, 68, 70, 129, 273, 277]). Generally, inertial metrics measured at upper peripheral locations (e.g., forearms and wrists) are not well suited for detecting gross motor tasks [130].

Algorithms may also combine inertial data with physiological metrics (e.g., ECG, heart rate, respiration rate, or skin temperature [107, 119, 192, 205, 206]). Physiological data can increase recognition accuracy by improving context. For example, adding heart rate discriminated between intensity levels (e.g., *running* and *running with weights* [192]). Lara and Labrador [147] demonstrated that physiological data can improve recognition accuracy by means of structural feature extraction [198]. However, physiological metrics may disrupt real-time task recognition, because they are not sensitive to sudden changes in physical activity. Further, adding heart rate did not improve activity recognition [258], because heart rate remains high after performing physically demanding activities (e.g., running), even when the individual was lying or sitting. HRV may overcome this limitation. The sensitivity of HRV decreases when < 30 seconds or > 2 mins of data is used for classification [85]. These algorithms extract time- and frequency-domain features and employ conventional classifiers (e.g., SVM [107, 205], Decision Trees [206, 258], RF [10, 192], Logistic

Table 2.6: Gross motor task recognition algorithms evaluation overview by Sensitivity (Sens.), Suitability (Suit.), Generalizability (Genr.), Composite Factor (Comp.), Concurrency (Conc.), and Anomaly Awareness (Anom.). Sensitivity is classified as Low ($\bigvee$), Medium ($\prod$), or High ($\bigwedge$), while other criteria are classified as conforming (C), non-conforming (NC), or requiring additional evidence (RE).

Category / Algorithm	Paper	Sens.	Suit.	Genr.	Comp.	Conc.	Anom.
Classical Machine Learning							
Artificial neural network	[119]	$\bigwedge$	C	NC	NC	NC	NC
Decision trees	[206]	$\bigwedge$	C	C	NC	NC	NC
	[258]	$\bigvee$	C	NC	NC	NC	NC
Ensemble	[192]	$\bigwedge$	C	C	NC	NC	NC
k-Nearest Neighbors	[130]	$\bigwedge$	NC	C	NC	NC	NC
	[137]	$\bigwedge$	NC	RE	NC	NC	NC
Logistic regression	[147]	$\bigwedge$	C	NC	NC	NC	NC
Plurality voting	[224]	$\bigwedge$	C	NC	NC	NC	NC
RF	[10]	$\bigwedge$	C	C	NC	NC	NC
Recurrent neural network	[22]	$\bigwedge$	NC	NC	NC	NC	NC
Relevance vector machines	[107]	$\bigwedge$	C	NC	NC	NC	NC
SVM	[205]	$\bigwedge$	C	C	NC	NC	NC
	[131]	$\bigwedge$	NC	C	NC	NC	NC
	[245]	$\bigvee$	NC	NC	NC	NC	NC
Deep Learning							
CNN	[43]	$\bigwedge$	C	C	NC	NC	NC
	[12]	$\bigwedge$	C	NC	NC	NC	NC
	[95]	$\bigwedge$	C	C	NC	NC	NC
	[151]	$\bigwedge$	C	NC	NC	NC	NC
LSTM	[96]	$\bigwedge$	C	NC	NC	NC	NC
CNN + LSTM	[57]	$\bigwedge$	NC	RE	NC	NC	NC
	[212]	$\bigwedge$	C	NC	C	NC	NC
	[40]	$\prod$	C	NC	C	NC	NC
CNN + Gated Recurrent	[276]	$\bigwedge$	C	NC	C	NC	NC
Transformer	[53]	$\bigwedge$	C	NC	NC	C	NC
Probabilistic Graphical Model							
Bayesian network	[288]	$\bigvee$	C	NC	C	C	NC
Conditional random field	[94]	$\bigwedge$	NC	RE	NC	NC	NC
Gaussian mixture model	[261]	$\prod$	NC	NC	NC	NC	NC
Hidden Markov model	[104]	$\bigwedge$	NC	RE	NC	NC	NC
	[125]	$\prod$	NC	NC	NC	NC	NC
Knowledge-driven							
Dynamic time warping	[52]	$\bigwedge$	NC	RE	NC	NC	NC
Principal component analysis	[209]	$\bigwedge$	C	NC	NC	NC	NC
Trigger-based	[200]	$\bigwedge$	NC	RE	NC	C	NC

Regression [147]).

Classical machine learning algorithms involving vision-based metrics leverage optical flow extracted from stationary cameras for gross motor task recognition (e.g., [131, 137]). Activity specific motion descriptors derived from optical flow are used as features to train a machine learning classifier (e.g., SVM [131] or k-Nearest Neighbors [137]). The algorithms are typically evaluated using the publicly available *Weizmann* [74] or *KTH Actions* datasets [245].

Generally, classical machine learning-based gross motor task detection algorithms typically have high *sensitivity*, primarily due to the atomic and repetitive nature of gross motor tasks. These algorithms conform with *suitability* when the metrics incorporated are wearable and reliable [10, 107, 119, 147, 192, 205, 206, 224, 258], and are non-conforming otherwise [130, 131, 137, 245]. Overall, the algorithms conform with *generalizability*, as they typically achieved high accuracy using a leave-one-subject-out cross-validation [10, 130, 131, 192, 205, 206]. All evaluated algorithms are non-conforming for the *concurrency*, *composite factor*, and *anomaly awareness* criteria.

2.3.7.2 Deep Learning Methods

Classical machine learning algorithms require handcrafted features that are highly problem-specific, and generalize poorly across task categories [230]. Additionally, those algorithms cannot represent the composite relationships among atomic tasks, and require significant human effort to select features and sensor data thresholding [230]. Comparative studies indicate deep learning algorithms outperform classical machine learning when large amount of training data is available [71, 230, 248].

Deep learning algorithms involving inertial metrics typically require little to no sensor data preprocessing. A CNN detected eight gross motor tasks (e.g., falling, running, jumping, walking, ascending, and descending a staircase) using raw acceleration data [43]. Although inertial data preprocessing is not required, it may be advantageous in some situations. For example, a CNN algorithm transformed the x, y, and z acceleration into vector magnitude data in order to minimize the acceleration's rotational interference [151]. The acceleration signal's *spectrogram*, which is a three dimensional representation of changes in the acceleration signal's energy as a function of frequency and time, was used to train a CNN model [12]. Employing the spectrogram improved the classification accuracy and reduced the computational complexity significantly [12]. Other deep learning algorithms

combine inertial data with physiological metrics to improve the classifier's effectiveness (e.g., ECG, and photoplethysmogram [178]).

Most recent algorithms leverage publicly available huge benchmark datasets (e.g., [227, 229]) to build deeper and more complex task recognition models. Deep learning algorithms combine CNNs with sequential modeling networks (e.g., LSTM [40, 212], Gated Recurrent Units [276]) to detect composite gross motor tasks from inertial data. The *DEBONAIR* algorithm [40] incorporated multiple convolutional sub-networks to extract features based on the input metrics' dynamicity and passed the sub-networks' feature maps to LSTM networks to detect composite gross motor tasks (e.g., vacuuming, nordic walking, and rope jumping). The *AROMA* algorithm [212] recognized atomic and composite tasks jointly by adopting a CNN + LSTM architecture, while *InnoHAR* algorithm [276] combined the *Inception* CNN module with Gated Recurrent Units to detect composite gross motor tasks. Several other algorithms draw inspiration from natural language processing to detect gross motor task transitions [259] and concurrency [53] by utilizing bi-directional LSTMs and Transformers, respectively. Bi-directional LSTMs concatenate information from positive as well as negative time directions in order to predict tasks, whereas Transformers incorporate self-attention mechanisms to draw long-term dependencies by focusing on the most relevant parts of the input sequence.

RFID indoor localization is common for task recognition (e.g., [52, 57, 94]). The RFID's *received signal strength indicator* and *phase angle* metrics are used to determine the relative distance and orientation of the tags with respect to the associated embedded environment readers [242]. The two common task identification methods are: i) tag-attached, and ii) tag-free [57]. *DeepTag* [57] introduced an advanced RFID-based task recognition algorithm that identified tasks in both tag-attached and tag-free scenarios. The deep learning-based algorithm used a preprocessed *received signal strength indicator* and *phase angle* information that combined a CNN with LSTMs in order to predict seven ADL tasks. Generally, the gross motor task recognition algorithms involving indoor localization have high *sensitivity*, but do not conform with *suitability* and *composite factor*.

Deep learning algorithms' increased network complexity and abstraction alleviates most of the classical machine learning algorithms' limitation, resulting in high sensitivity, especially when the data is abundant [71, 230]; however, caution must be exercised to not overfit the algorithms. Deep learning algorithms can achieve high classification accuracy on multi-modal sensor data without requiring special feature engineering for each modality. For example, a hybrid deep learning algorithm trained using an 8-channel sEMG and

inertial data detected thirty gym exercises (e.g., dips, bench press, rowing) [62]. Deep learning algorithms rarely validate their results via leave-one-subject-out cross-validation, as in most cases the algorithms are validated by splitting all the available data randomly into training and validation datasets; therefore, the algorithms' *generalizability* criteria either requires additional evidence, or is non-conforming.

2.3.7.3 Probabilistic Graphical Models

Algorithms' task predictions are not always accurate, as there is always some uncertainty associated with the predictions, especially when tasks overlap with one another, or share similar motion patterns (e.g., running vs. running with weights). Additionally, humans may perform two or more tasks simultaneously, which complicates task identification when using classical and deep learning methods that are typically trained to predict only one task occurring at a time. Probabilistic graphical task recognition algorithms are adept at managing these uncertainties, and have the ability to model simultaneous tasks.

Probabilistic graphical models can detect gross motor tasks across various metrics (e.g., indoor localization [94], sEMG [261], inertial [122, 152], human-body pose [104], optical flow [125], and object detection [288]). Hidden Markov Models are the most widely utilized probabilistic graphical algorithm for gross motor task recognition (e.g., [104, 122, 125, 152]), because Hidden Markov Model's sequence modeling properties can be exploited for continuous task recognition [122]. Hidden Markov Models also allow for modeling the tasks hierarchically [152], and can distinguish tasks with intra-class variances and inter-class similarities [122]. Other probabilistic models (e.g., Gaussian Mixture Models [261]) can also detect gross motor tasks. A probabilistic graphical model, the *Interval-temporal Bayesian Network*, unified Bayesian network's probabilistic representation with interval algebra's [11] ability to represent temporal relationships between atomic events [288] to detect composite and concurrent gross motor tasks. The algorithm's *sensitivity* and *generalizability* are low and non-conforming, respectively. The algorithm's *suitability* is non-conforming, as it employed vision-based metrics. Finally, the algorithm's *composite factor* and *concurrency* conform.

2.3.7.4 Knowledge-Driven Algorithms

Gross motor rule-based task recognition algorithms incorporate template matching or thresholding to recognize tasks. A *Dynamic Time Warping* [235] based algorithm detected free-weight exercises by computing the similarity between Doppler shift profiles of the reflected RFID signals [52]. A principal component analysis thresholding algorithm detected ambulatory task transitions by analyzing the motion artifacts in ECG data induced by body movements [208, 209].

Rule-based algorithms can detect concurrent tasks, if the rules are relatively simple to derive using the sensor data. A multiagent algorithm [200] detected up to seven gross motor atomic tasks (e.g., dressing, cleaning, and food preparation). The algorithm detected up to two concurrent tasks using environmentally-embedded proximity sensors.

Rule-based systems are ideal for gross motor task detection when the sensor data is limited and can be comprehended in a relatively straightforward manner. For example, the prior rule-based multiagent algorithm detected concurrent tasks, as it was easy to form the rules using the proximity sensor data. Rule-based algorithms are unsuitable when the sensor data cannot be interpreted easily (i.e, instances of high dimensionality), or when there are a large number of tasks that have intricate relationships.

2.3.7.5 Discussion

Most machine learning based algorithms can detect gross motor tasks reliably with acceptable suitability and generalizability when the tasks are atomic and non-concurrent with repetitive motions (e.g., [10, 43, 205, 206]). The human-robot teaming domain often involves composite tasks that may occur concurrently. None of the existing gross motor task detection algorithms satisfy all the required criteria for the intended domain.

The interval-temporal algorithm [288] is the preferred approach for gross motor task detection. The algorithm can detect concurrent and composite tasks, but had low sensitivity and is non-conforming for suitability and generalizability, which can be attributed to the vision-based metrics and low-level Bayesian network's poor classification accuracy. However, the algorithm is independent of the metrics [288], as it operates hierarchically, utilizing the low-level atomic event predictions. Therefore, a modified version more suited to the intended domain may incorporate a classical machine learning algorithm (e.g., RF [10]) or a deep network (e.g., CNN [43]), depending on the amount of data available, to

detect the low-level atomic tasks using inertial metrics. The interval-temporal algorithm can be used to detect the composite and concurrent gross motor tasks.

2.3.8 Fine-Grained Motor Tasks

Fine-grained motor tasks often involve highly articulated and dexterous motions that can be performed in multiple ways. The execution and the time taken to complete the tasks differ from one human to the other. These aspects of fine-grained motor tasks can create ambiguity in the sensor data, making it difficult for the algorithms to detect such tasks; therefore, a wide range of methods adopting various sensing modalities exist for detecting fine-grained tasks accurately. The evaluation criteria for each reviewed fine-grained task recognition algorithm by algorithm category is provided in Table 2.7.

2.3.8.1 Classical Machine Learning

Classical machine learning algorithms are suitable for detecting fine-grained motor tasks only when the tasks are short in duration, atomic, or repetitive [88]. Among the classical machine learning algorithms, k-Nearest Neighbors (e.g., [127, 130, 137]), RF (e.g., [88, 156, 269]), and SVM (e.g., [176, 185, 216]) are the most popular choices for fine-grained motor task detection.

Several classical machine learning algorithms use egocentric wearable camera videos for detecting ADL tasks (e.g., [66, 176, 216]). Image processing techniques (e.g., histogram of orientation or spatial pyramids) are used to detect objects and conventional machine learning algorithms recognize the tasks from the detected objects. These algorithms may also incorporate saliency detectors [176], or depth information [66]) to identify the objects being manipulated. Temporal motion descriptive features from optical flow can also be used for recognizing fine-grained tasks. A k-Nearest Neighbors algorithm classified the fine-grained motor tasks [137] based on a histogram constructed using the motion descriptors from optical flow.

Forearm sEMG signals can detect tasks that are difficult for a vision-based algorithm to differentiate when using the same conventional classifiers. A comparison between an sEMG (i.e., Myo armband [239]) and a motion capture sensor revealed that the former had higher efficacy in recognizing fine-grained motions (e.g., grasps and assembly part manipulation tasks) [130]. The classifiers with the sEMG data detected the minute variation in the

Table 2.7: Fine-grained motor task recognition algorithms' evaluation overview.

Category / Algorithm	Paper	Sens.	Suit.	Genr.	Comp.	Conc.	Anom.
Classical Machine Learning							
Ensemble	[185]	$\bigwedge$	C	C	C	NC	NC
k-Nearest Neighbors	[127]	$\bigwedge$	C	C	NC	NC	NC
	[130]	$\bigvee$	NC	NC	NC	NC	NC
	[137]	$\bigvee$	NC	NC	NC	NC	NC
RF	[269]	$\bigwedge$	NC	C	NC	NC	NC
	[156]	$\bigwedge$	NC	RE	C	NC	NC
	[88]	$\bigvee$	NC	NC	C	NC	NC
SVM	[143]	$\bigwedge$	NC	C	NC	NC	NC
	[216]	$\bigvee$	NC	NC	NC	NC	NC
	[176]	$\bigvee$	NC	NC	NC	NC	NC
	[287]	$\bigvee$	NC	NC	NC	NC	NC
Deep Learning							
CNN	[142]	$\bigwedge$	C	C	NC	NC	C
	[157]	$\bigwedge$	NC	RE	C	NC	NC
	[154]	$\bigvee$	NC	NC	C	NC	NC
	[171]	$\prod$	NC	NC	C	NC	NC
	[34]	$\bigwedge$	NC	NC	NC	NC	NC
CNN + LSTM	[264]	$\bigwedge$	NC	NC	NC	NC	NC
	[62]	$\bigvee$	NC	NC	NC	NC	NC
LSTM	[66]	$\bigvee$	NC	NC	NC	NC	NC
LSTM bi-directional	[291]	$\bigwedge$	NC	C	C	C	NC
Residual + Attention	[179]	$\bigwedge$	C	NC	C	NC	NC
	[8]	$\prod$	C	NC	C	NC	NC
Transformer	[289]	$\bigwedge$	C	C	NC	NC	NC
Probabilistic Graphical Model							
Bayesian network	[165]	$\bigwedge$	C	RE	C	C	NC
	[274]	$\bigwedge$	NC	NC	NC	NC	NC
	[60]	$\bigwedge$	NC	RE	NC	NC	NC
	[92]	$\bigvee$	C	NC	NC	NC	NC
Conditional random field	[58]	$\bigvee$	NC	NC	C	NC	NC
Gaussian mixture model	[183]	$\bigwedge$	C	C	NC	NC	NC
	[184]	$\bigwedge$	C	C	NC	NC	C
Hierarchical latent SVM	[163]	$\bigwedge$	NC	C	C	C	NC
Temporal memory	[282]	$\bigwedge$	C	C	NC	NC	NC
Markov chains	[232]	$\bigwedge$	NC	NC	C	C	NC
Probabilistic Neural	[272]	$\bigwedge$	C	NC	NC	NC	NC
Temporal graph	[161]	$\bigwedge$	C	RE	C	C	NC

muscle associated with each grasp, resulting in significantly higher recognition accuracy than using the motion capture data.

Some classical machine learning algorithms that use a single IMU can classify fine-grained motor ADL tasks (e.g., eating and drinking [282]), and assembly line activities (e.g., hammering and tightening screws [127]). This approach is suitable for tasks involving a single hand (i.e., the dominant) when the number of recognized tasks is small (e.g., < 5). For instance, five assembly line tasks were recognized using a wrist worn IMU's acceleration and angular velocity data [127]. The associated time- and frequency-domain features were used to train a k-Nearest Neighbors algorithm to classify the tasks. A two-stage classification approach using acceleration metrics obtained by a wrist-worn accelerometer recognized eating and drinking [282]. However, when the tasks are composite or larger in number, the algorithms augment the IMU with different sensing modalities. Algorithms typically combine IMU with sEMG metrics measured at upper peripheral locations (e.g., the forearms and wrists) in order to capture highly articulated motions [130]. Increasing sensing modalities provides more task context, enabling an algorithm to discriminate a broader set of tasks.

A system attempted to recognize twenty-three composite clinical procedures by using metrics from two Myo armbands and statically embedded cameras [88]. The Myo's sEMG and inertial metrics were combined with the camera's human body pose metric to train a RF classifier with majority voting. Many clinical procedures require multiple articulated fine-grained motions that range from < 10 seconds (s) to $> 60s$ to complete. Long-duration fine-grained procedures are difficult to detect due to intra-class variability, inter-class similarity, and individual differences among participants. The video provided contextual information that improved the procedure recognition accuracy by alleviating intra-class variance and inter-class similarity.

A multi-modal framework, incorporating five inertial sensors, data gloves, and a bio-signal sensor detected eleven atomic tasks (e.g., writing, brushing, typing) and eight composite tasks (e.g., exercising, working, meeting) [185]. A hybrid ensemble approach combined classifier selection and output fusion. The sensors' inputs were initially recognized by a Naive Bayes selection module. The selection module's task probabilities chose a set of task-specific SVM classifiers that fused their predictions into a matrix in order to identify the tasks.

Classical machine learning algorithms' classification accuracies range between 45% - 65%; thus, they generally have low *sensitivity*. The algorithms' *suitability* criterion depend

on the metrics employed. The *composite factor* and *generalizability* criteria also vary across algorithms, as they depend on the tasks detected and the validation methodology. Overall, most algorithms are non-conforming for the *concurrency* and *composite* factors, making them unsuitable for detecting fine-grained motor tasks for the intended HRT domain.

2.3.8.2 Deep Learning

The ambiguous, convoluted sensor data from fine-grained motor tasks causes the feature engineering and extraction to be laborious. Deep learning algorithms overcome this limitation by automating the feature extraction process. There are three different types of deep learning algorithms for fine-grained motor task recognition: i) Convolutional, ii) Recurrent, and iii) Hybrid. Convolutional algorithms typically incorporate only CNNs to learn the spatial features from sensor data for each task and distinguish them by comparing the spatial patterns (e.g., [34, 142, 154, 157, 171]). Recurrent algorithms detect the tasks by capturing the sequential information present in the sensor data, typically using memory cells (e.g., [66, 291]). Hybrid algorithms extract spatial features and learn the temporal relationships simultaneously by combining convolutional and recurrent networks [57, 62, 264].

Deep learning algorithms using egocentric videos from wearable cameras combine object detection with task recognition. A CNN with a late fusion ensemble predicted the tasks from a chest-mounted wearable camera [34] by incorporating relevant contextual information (e.g., time and day of the week) to boost the classification accuracy. Two separate CNNs were combined together to recognize objects of interest and hand motions [171]. The networks were fine tuned jointly using a triplet loss function to recognize fine-grained ADL tasks with medium to high *sensitivity*.

Analyzing changes in body poses spatially and temporally can provide important cues for fine-grained motor task recognition [163]. An end-to-end CNN network exploited camera images for estimating fifteen upper body joint positions [193]. The estimated joint positions permitted discriminating features to recognize tasks. The CNN architecture had two levels: i) fully-convolutional layers that extracted the salient feature, or heat maps, and ii) fusion layers that learned the spatial dependencies between the joints by concatenating the convolutional layers. The CNN-estimated joint positions served as input to train a multi-class SVM that predicted twelve ADL tasks with high *sensitivity*.

Hybrid deep learning algorithms are becoming increasingly popular for task recognition

across metrics [57, 62, 264]. An optical flow-based algorithm [264] leveraged deep learning to extract temporal optical flow features from the salient frames, and incorporated a multilayer LSTM to predict the tasks using the temporal optical flow features. Another hybrid deep learning algorithm [62] trained on the sEMG and inertial metrics detected assembly tasks. The algorithm's CNN layers extracted spatial features from the merics at each timestep, while the LSTM layers learned how the spatial features evolved temporally. Hybrid algorithms can provide excellent expressive and predictive capabilities; however, these algorithms' performance relies heavily on the size of the training dataset [62]. Other hybrid deep learning algorithms combine IMU with spectrogram metrics to improve the classifier's effectiveness (e.g., [160, 187]).

A CNN-based algorithm incorporated inertial metrics from an off-the-shelf smartwatch to detect twenty-five atomic tasks (e.g., operating a drill, cutting paper, and writing) [142]. A Fourier transform was applied to the acceleration data to obtain the corresponding spectrograms. The CNN identified the spatial-temporal relationships encoded in the spectrograms by generating distinctive activation patterns for each task. The algorithm also rejected (i.e., detected) unknown instances.

Deep learning algorithms can recognize concurrent and composite fine-grained motor tasks directly from raw sensor data using complex network architectures, provided sufficient data is available [199, 291]. Human task trajectories are continuous in that the current task depends on both past and future information. A deep residual bidirectional LSTM algorithm [291] detected the *Opportunity* dataset's composite tasks by incorporating information from positive as well as negative time directions. The dataset contains five composite ADLs (e.g., relaxation, preparing coffee, preparing breakfast, grooming, cleaning) involving a total number of 211 atomic events (e.g., walking, sitting, lying, opening doors, reaching for an object). Several metrics, including acceleration and orientation of various body parts, and three-dimensional indoor position were gathered.

Recent deep learning algorithms leverage attention mechanisms to model long-term dependencies from inertial data [8, 179, 219, 289]. The *ResNet-SE* algorithm [179] classified composite fine-grained motor tasks on three publicly available datasets. The algorithm incorporated residual networks to address loss degradation, followed by a squeeze-and-excite attention function to modulate the relevance of each residual feature map. The *Multi-ResAtt* algorithm [8] incorporated residual networks to process inertial metrics from IMUs distributed over different body locations, followed by bidirectional Gated Recurrent Units with attention mechanism to learn time-series features.

Generally, deep learning algorithms are highly effective at detecting atomic fine-grained motor tasks, but their ability to detect composite and concurrent tasks reliably is indeterminate. The latter may be due to insufficient ecologically-valid concurrent, composite task recognition datasets available publicly. Utilizing generative adversarial networks [158, 270] to expand datasets by producing synthetic sensor data may alleviate the issue.

2.3.8.3 Probabilistic Graphical Models

Bayesian networks are the most common probabilistic graphical models for fine-grained motor task detection (e.g., [60, 92, 165, 274]), followed by Gaussian Mixture Models (e.g., [183, 184]). Many such algorithms augment the inertial data with a different sensing modality (e.g., [92, 183, 184, 274]) in order to provide more task context, which enables discriminating a broader set of tasks. Recognition of up to three day-to-day early morning tasks [183] augmented with a microphone, resulted in the recognition of six tasks [184]. The intended HRT domain requires multiple sensors to detect tasks belonging to different activity components, although adding new modalities arbitrarily may deteriorate the classifier performance [62].

Hierarchical graphical models detect composite tasks by decomposing them into a set of smaller classification problems. Fathi et al.'s [58] meal preparation task detection algorithm decomposed hand manipulations into numerous atomic actions, and learned tasks from a hierarchical action sequence using conditional random fields. Another hierarchical model that operated at three levels of abstraction detected concurrent, composite tasks using body poses [163].

Identifying the causality (i.e., action and reaction pair) between two events allows for easier human interpretation, and for modeling far more intricate temporal relationships [161]. A graphical algorithm incorporated the Granger-causality [76, 77] test for uncovering cause-effect relationships among atomic events [161]. The algorithm employed a generic Bayesian Network to detect the atomic events. A temporal causal graph was generated via the Granger-causality test between atomic events. Each graph represented a particular task instance. The graph nodes represented the atomic events and directed links with weights represented the cause-effect relationships between the atomic events. An artificial neural network is trained using these graphs as inputs to predict the concurrent, composite tasks. The algorithm was evaluated on the *Opportunity* [229] and *OSUPEL* [29] datasets, indicating that the algorithm is independent of the metrics.

Overall, probabilistic graphical models typically have high *sensitivity* for detecting fine-grained motor tasks. The algorithms, especially hierarchical (e.g., [163]) and the Granger-causality based temporal graph [161], are independent of the metrics due to data abstraction; therefore, their *suitability* is classified as conforming. Most task recognition algorithms are susceptible to individual differences (see Table 2.7). Even those that conform with generalizability may experience a significant decrease in accuracy when classifying an unknown human's data [142]; thus, the *generalizability* criterion requires additional evidence. Algorithms can only identify tasks reliably for humans on which they were trained, suggesting that online and self-learning mechanisms are needed to accommodate new humans [272]. The *composite factor* and *concurrency* vary across algorithms, but are conforming overall. The *anomaly awareness* criterion is classified as non-conforming, as most probabilistic graphical models do not detect out-of-class tasks.

2.3.8.4 Discussion

Classical machine learning algorithms are unreliable for detecting fine-grained motor tasks due to poor sensitivity and generalizability. Deep learning algorithms can detect the atomic fine-grained motor tasks reliably, but not concurrent, composite tasks. Moreover, deep learning typically requires a large number of parameters, very large datasets and can be difficult to train [163]. Deep learning's automatic feature learning capability prohibits exploiting explicit relationships among tasks and semantic knowledge, making it difficult to detect concurrent, composite fine-grained motor tasks. Probabilistic graphical models offer some suitable alternatives; however, none of the existing algorithms satisfy all the required criteria for the intended domain.

The Granger-causality based temporal graph algorithm [161] and the three-level hierarchical algorithm [163] are the most suitable for fine-grained motor task detection given all the other algorithms. Both algorithms have high sensitivity and can detect concurrent and composite tasks. The Granger-causality algorithm conforms with suitability, but requires additional evidence to substantiate its generalizability. The hierarchical algorithm conforms with generalizability, but is non-conforming with suitability, as it employed a vision-based system for estimating human-body pose metric. However, the metric can be estimated using a series of inertial motion trackers [56]; therefore, a human-robot teaming domain friendly version of both algorithms can be developed theoretically.

2.3.9 Tactile Tasks

Tactile interaction occurs when humans interact with objects around them (e.g., keyboard typing, mouse-clicking and finger gestures). Individual classifications for each tactile task algorithm by its category are provided in Table 2.8.

Table 2.8: Tactile task recognition algorithms' evaluation overview.

Category / Algorithm	Paper	Sens.	Suit.	Genr.	Comp.	Conc.	Anom.
Classical Machine Learning							
Decision trees	[109]	⋀	NC	C	NC	NC	NC
Ensemble	[35]	⋀	C	C	NC	NC	NC
SVM	[93]	⋀	NC	RE	NC	NC	NC
Voting	[247]	⋀	NC	NC	NC	NC	NC
Deep Learning							
CNN	[48]	⋀	NC	NC	NC	NC	NC
CNN + LSTM	[223]	⋀	NC	NC	NC	NC	NC
Probabilistic Graphical Model							
Gaussian mixture model	[111]	⋀	NC	C	NC	NC	NC
Hidden Markov model	[285]	⋀	NC	NC	NC	NC	NC
Naive Bayes classifier	[42]	⋀	NC	NC	NC	NC	NC
	[41]	⋀	NC	NC	NC	NC	NC

2.3.9.1 Classical Machine Learning

Most classical machine learning algorithms incorporate inertial metrics measured at the fingers or dorsal side of the hand. These approaches typically detect finger gestures and keystrokes depending on the measurement site (e.g., [35, 109, 166, 247]). Several of these approaches employ multiple ring-like accelerometer devices worn on the fingers (e.g., [109, 247, 292]). The time- and frequency-domain features (e.g., minimum, maximum, standard deviation, energy, and entropy) extracted from the acceleration signals were used to train classical machine learning algorithms (e.g., decision tree classifier and majority voting) to detect finger gestures (e.g., finger rotation and bending) and keystrokes. Although these approaches incorporated inertial metrics, none conform with *suitability* due to lack

of reproducibility (i.e., the ring-like sensor is not commercially available) and wearing a ring-like device may hinder humans' dexterity, impacting task performance negatively.

Inertial metrics from the dorsal side of the hand detected seven office tasks (e.g., keyboard typing, mouse-clicking, writing) [35]. Time- and frequency-domain features extracted from the acceleration signals were used to train an ensemble classifier. The algorithm achieved high accuracy ($> 90\%$) in an in-the-wild evaluation. Most misclassifications occurred during transitions between tasks, implying that inertial-based tactile task recognition may be susceptible to task transitions due to signal variations. The high error rates during transitions can lead to lower classification accuracy, especially when tasks switch frequently.

Classical machine learning algorithms' generally have high *sensitivity*. The algorithms' are typically non-conforming for the *suitability* criterion, as many supporting research efforts focus on developing and validating new sensor technology for sensing tactility, rather than detecting tactile tasks (e.g., [101, 113, 201]). The *generalizability* criteria also vary across algorithms, as they depend on the validation methodology. Finally, the algorithms are non-conforming for the *concurrency* and *composite* factors, making them unsuitable for detecting tactile tasks for the intended HRT domain.

2.3.9.2 Deep Learning

Several publicly available sEMG-based hand gesture datasets (e.g., [13, 16, 112]) support deep learning algorithms to detect tactile hand gestures (e.g., [48, 223]). A hybrid deep learning model consisting of two parallel paths (i.e., one LSTM path and one CNN path) was developed [223]. A fully connected multilayer fusion network combined the outputs of the two paths to classify the hand gestures.

Recognizing tactile tasks is an under-developed area of research, as the tasks are nuanced and often overshadowed by fine-grained motor tasks. Generally, deep learning algorithms have high *sensitivity*; however, the incorporated sEMG metrics with a random dataset split for validation cause them to not conform with the *suitability* and *generalizability* criteria.

2.3.9.3 Probabilistic Graphical Model

Probabilistic graphical models for tactile task recognition typically involve simple algorithms (e.g., Hidden Markov Models [285], Gaussian Mixture Models [111], and Bayesian Networks [41, 42]) when compared to the prior gross motor and fine-grained motor chapters (see Chapters 2.3.7 and 2.3.8), as the tasks detected are inherently atomic (e.g., hand and finger gestures). sEMG signals are one of the most frequently used metrics for detecting hand and finger gestures (e.g., [42, 48, 223, 285]). Chen et al.'s [42] gesture recognition algorithm pioneered the use of sEMG signals. Twenty-five hand gestures (i.e., six wrist actions and seventeen finger gestures) were detected using a 2-channel sEMG placed on the forearm. A Bayesian classifier was trained using the mean absolute value and autoregressive model coefficients extracted from the sEMG. The algorithm was extended to include two accelerometers, one placed on the wrist and the other placed on the dorsal side of the hand [41].

Overall, probabilistic graphical models tend to have high *sensitivity* for detecting tactile tasks. Most algorithms are susceptible to individual differences and incorporate sEMG metrics; thus, the algorithms are non-conforming for the *suitability* and *generalizability* criteria. Additionally, the algorithms are non-conforming for the *concurrency* and *composite* factors, as the evaluated tactile tasks are inherently atomic.

2.3.9.4 Discussion

All data-driven algorithms can detect tactile tasks with $> 80\%$ accuracy [35, 41, 93, 109, 111, 223, 247, 285] primarily because the detected tasks (i.e., finger and hand gestures) were atomic; therefore, the algorithms have high *sensitivity*. Except for the office-based tactile task classifier [35] none of the existing algorithms conform with *suitability*, because either the sensors incorporated were commercially unavailable for reproducibility, or the metrics employed were unreliable. All the algorithms are non-conforming with the *concurrency* and *composite factor* criteria, because tactile tasks are rarely composite or concurrent. Finally, none of the algorithms detect out-of-class instances; therefore, they do not conform with *anomaly awareness*. A recommended tactile task detection algorithm to support the intended domain is the interval-temporal algorithm [288], or the Granger-causality based temporal graph [161] with the inclusion of inertial metrics measured at the dorsal side of the hand to capture the tactile component, along with the fine-grained motor component.

2.3.10 Summary

The goal is to develop a recognition algorithm that can identify tasks performed by HRTs working in unstructured, dynamic environments. HRTs often perform a wide range of tasks, such that the set of all tasks performed by human teammates may involve combinations of all activity components. Thus, an overall task recognition model capable of detecting tasks with multiple different activity components is desired. None of the reviewed algorithms meet all the criteria necessary to achieve the goal, due to identifying tasks with a limited set of activity components and the algorithms' limitations with regard to the evaluation criteria: sensitivity, suitability, generalizability, composite factor, concurrency, and anomaly awareness.

Two algorithms, Grana et al.'s [75] and Ishimaru et al.'s [97], come the closest by identifying four activity components. The former identified a gross motor task, while the latter identified a fine-grained motor task in addition to detecting visual, cognitive, and auditory tasks. Both algorithms failed to satisfy all the required evaluation criteria. Most other algorithms detected tasks involving at most two activity components: gross and fine-grained motor, fine-grained motor and tactile, or visual and cognitive tasks. None of the reviewed algorithms detected tasks across all seven activity components.

Several algorithms conform with sensitivity, suitability, and generalizability. Other than the cognitive and speech task recognition algorithms, there exists at least one algorithm that can detect tasks reliably while conforming with suitability and generalizability for each individual activity component. Although, suitable alternatives were identified for both the cognitive and speech components (see Chapters 2.3.3.3 and 2.3.4.2). Existing algorithms are highly limited in satisfying the other three criteria: concurrent, composite, and anomaly awareness.

The ability to recognize the human's composite tasks is crucial for the intended domain. Thirteen algorithms, all of which were either gross or fine-grained motor task recognition algorithms, attempted to detect composite tasks. Eight algorithms detected tasks with high accuracy, while only three [161, 165, 185] managed to detect tasks with high accuracy using reliable metrics from wearable sensors. A similar trend was observed for concurrency detection. Eight algorithms attempted to detect concurrent tasks, seven of which belonged to the gross or fine-grained motor categories. Five of those algorithms detected tasks with high sensitivity, while only two [161, 165] satisfied the sensitivity and suitability criteria.

Only two algorithms [142, 184] conform with anomaly awareness. The remaining al-

gorithms' primary limitation for anomaly awareness was the assumption that humans will not perform tasks outside the predefined set, which is not the case for the intended domain. Therefore, there remains a need for a task recognition algorithm that can detect out-of-class instances reliably. Such an algorithm can draw from existing novelty and anomaly detection research (e.g., [37, 136, 175, 203, 214]). This dissertation focuses on identifying tasks performed by HRTs working in unstructured, dynamic environments. The developed multi-dimensional task recognition algorithm detected concurrent, composite tasks across all activity components using reliable metrics obtained from unobtrusive wearable sensors.

Chapter 3: Multi-Dimensional Task Recognition Algorithm

An adaptive human-robot teaming system needs an algorithm capable of detecting the tasks performed by the human teammates in order to adapt interactions and autonomy levels intelligently. Existing task recognition algorithms are not viable for an adaptive system, as they only detect tasks from a subset of activity components and rarely detect concurrent, composite tasks (see Chapter 2.3.10); thus, the algorithms are unable to provide the necessary information. Further, the adaptive system cannot rely on environmentally embedded sensors, as HRTs are often required to operate in dynamic, unstructured environments. This chapter introduces a multi-dimensional task recognition algorithm employing wearable sensors that in the future can be utilized by an adaptive human-robot teaming system.

3.1 Task Terminology

The tasks performed by HRTs can be classified into three categories hierarchically:

- *Atomic* tasks are singular, sometimes simple actions or activities that may last only for a short period of time. These tasks cannot be further subdivided and may consist of one or more activity components. These tasks represent the lowest level task in the hierarchical decomposition.

- A *composite* task aggregates multiple atomic tasks into a more complex task. These tasks represent the majority of mid-level tasks within a hierarchical decomposition. A composite task may encompass *sub-composite* tasks (i.e., a smaller composite task) if the composite task requires multiple series of coordinated tasks. For example, using a Walkie-Talkie is a composite task that can be subdivided into two sub-composite tasks: i) listening to the information, and ii) responding to the information, where each sub-composite task requires its own set of coordinated actions, and the latter sub-composite task's execution is contingent on several factors (e.g., the information's relevance to the human teammate, or the ongoing mission, or a combination of both).

- A *mission* task is a distinct assignment that forms an integral part of a broader goal or objective. For instance, removing debris to clear a roadblock is a mission task

that forms a vital part of post-tornado disaster response efforts. Mission tasks are often associated with the military, space exploration, emergency response, and other complex operations, where a series of coordinated atomic and composite tasks are required to complete the mission task successfully. These tasks represent the highest level task in a hierarchical decomposition.

HRTs often perform a wide variety of tasks during a mission, such that the set of all tasks (i.e., atomic and composite) performed by human teammates may involve combinations of all activity components. Thus, an overall task recognition algorithm capable of detecting atomic and composite tasks with multiple different activity components is desired. This dissertation presents a multi-dimensional task recognition algorithm that utilizes wearable sensors for identifying atomic and composite tasks that encompass multiple activity components performed by humans in unstructured, dynamic environments. It is also important to note that this dissertation focuses only on detecting atomic and composite tasks. Detecting mission tasks is outside the scope of this dissertation.

3.2 Algorithm Overview

An overview of the multi-dimensional task recognition architecture is provided in Figure 3.1. The task recognition algorithm hinges on incorporating reliable metrics obtained from wearable sensors. A selected set of task recognition metrics (see Chapter 3.3) obtained from the wearable sensors are filtered (see Chapter 3.4) before being fed into the multi-dimensional task recognition algorithm (see Chapter 3.5). The filtered metrics are channeled appropriately to the individual component task detection algorithms (i.e., cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile) as shown in Figure 3.1. These algorithms are developed independently (see Chapter 3.5) using their respective metrics to detect the corresponding atomic tasks contributing to the associated activity component. The *Fusion* algorithm (see Chapter 3.6) consolidates the predictions from the individual component algorithms in order to create a list of atomic tasks for the *Composite and Concurrent* task recognition algorithm. Additionally, the *Fusion* algorithm learns how the atomic task detections from different components are related to each other in order to account for prediction inconsistencies between the individual algorithms. The vector-encoded atomic task detections from the *Fusion* algorithm serve as input to the *Composite and Concurrent* task detection algorithm (see Chapter 3.7) that employs a

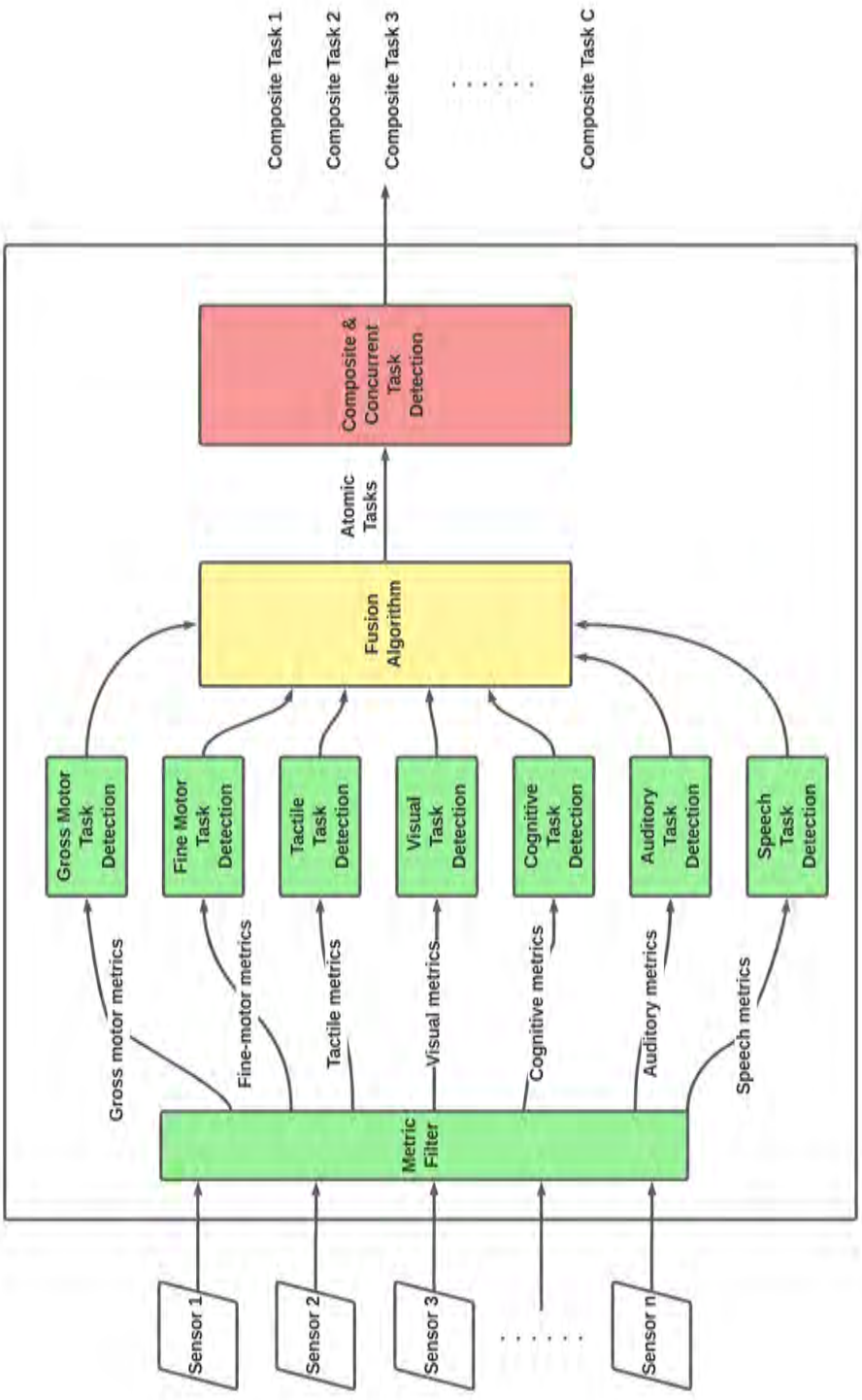


Figure 3.1: Multi-dimensional Task Recognition Architecture.

temporal convolutional network to detect the concurrent, composite tasks.

3.3 Metrics Selection

The task recognition metrics are chosen based on their sensitivity, versatility, and suitability (see Chapter 2.2), and the available sensors. The mapping between the sensors, associated metrics, and the task activity components are provided in Table 3.1.

Table 3.1: The wearable sensors and the corresponding metrics incorporated by the multi-die task recognition algorithm. NOTE: Grey cells represent the metric's association with the corresponding activity component.

Sensor	Metric	Cognitive	Speech	Audio	Visual	Gross motor	Fine-grained motor	Tactile
BioHarness	HRV							
	Heart rate							
	Respiration rate							
	Postural magnitude							
Pupil Core	Fixations							
	Saccades							
	Pupil dilation							
	Blink latency							
	Blink rate							
Microphone	Voice intensity							
	Voice pitch							
	Speech rate							
	MFCCs							
	Spectrogram							
Reed decibel meter	Noise level							
Xsens	Inertial							
Myo Armband	Inertial							
	sEMG							

The cognitive tasks are detected using the blink rate, blink latency, pupil dilation, and HRV metrics. The HRV metric is derived from Biopac's measured ECG signal.

the speech-reliant tasks are detected using the MFCCs and the speech-based metrics, while the auditory tasks are detected by combining the spectrogram and the noise level metrics. The noise level is acquired using a REED R8080 decibel meter, while the acoustic (i.e., spectrogram and MFCCs) and speech-based metrics (i.e., speech rate, pitch, and voice intensity) are calculated from a 44100 KHz dual-channel audio signal captured by a Shure PGX1 microphone. The chosen metrics are filtered and combined individually using separate task detection algorithms (see Chapter 3.5) for each activity component to recognize the atomic tasks.

The visual tasks are detected using fixations and saccades in conjunction with the head movement captured using the Xsens' forehead inertial data (as indicated in Figure 3.2).

The eye gaze metrics, fixations, and pupil dilations are measured via the Pupil Core and Neon eye trackers from *Pupil Labs*. The other eye gaze metrics (e.g., saccades, blink rate, and blink latency) are derived from the eye tracker's measured eye gaze signals.

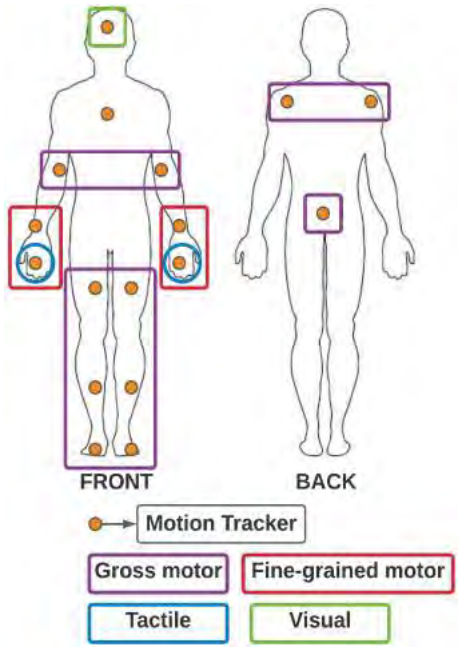


Figure 3.2: Xsens motion tracker locations

The gross motor tasks are detected by combining the inertial metrics measured at the lower and upper body positions (i.e., waist, shoulders, biceps, thighs, calves, and ankles) with the physiological metrics (i.e., heart rate, respiration rate, and postural magnitude). The Xsens MTw Awinda motion capture system [207] measures inertial data (i.e., acceleration and angular velocities) using seventeen IMU-based motion trackers worn at various body parts, as depicted in Figure 3.2. The physiological metrics are provided by the Biopac BioHarness™.

The Myo armband sensor measures the forearm sEMG and inertial data [239]. The tactile tasks are detected using the Xsens' inertial data measured at the dorsal side of the hand (see Figure 3.2) as well as the Myos' forearm sEMG data. The fine-grained motor task detection uses the Myos' forearm inertial data and the Xsens' inertial data measured on the wrists (highlighted in Figure 3.2) in addition to the metrics used for tactile task detection.

3.4 Metrics Filtering

The individual component task detection algorithms combine the respective metrics to detect the atomic tasks for each activity component. Noisy sensor readings may generate inaccurate detections; thus, metrics are filtered before being incorporated into the task detection algorithms.

The inertial, sEMG, heart rate, respiration rate, postural magnitude, and noise level metrics are smoothed using a moving average filter to improve the algorithms' generalizability and to remove or reduce unwanted signal artifacts. The HRV metric does not require any filtering.

The eye tracker's raw gaze data contains Gaussian noise with a slowly changing mean [197]. The high frequency noise is caused by inconsistent measurements, while the low frequency noise is due to a drift caused by pupil size changes and head movements [197]. The eye tracker implements a dispersion-based fixation detector [236] internally to convert the noisy raw gaze data into a series of fixations. The pupil dilation, blink rate and blink latency metrics does not require any filtering.

The microphone's audio stream is converted from a stereo audio signal into a mono audio signal. The acoustic and speech metrics are extracted by segmenting the mono signal into overlapping audio windows based on different window sizes. The segmented audio windows are decomposed with a short-time Fourier transform at 10 milliseconds intervals looking back over 25 milliseconds of Hann audio frames to yield a 100-length spectrogram [82, 91, 144, 159]. The resulting linear spectrogram is converted into a 64-bin log-scaled Mel spectrogram. The MFCCs are obtained by transforming the log-scaled Mel spectrogram using discrete cosine transformation.

The speech metrics (i.e., voice intensity, pitch, and speech rate) are derived using the metric extraction process employed in the real-time speech workload estimation algorithm [61, 87]. The metrics are calculated by decomposing the segmented windows into shorter *audio frames* and examining the *frames* at $10ms$ intervals looking back over $25ms$ of audio.

3.5 Individual Task Detection Algorithms

The individual component task detection algorithms combine the respective metrics described in Chapter 3.3 to detect the corresponding atomic tasks. Each individual task detection algorithm employs a different machine learning technique to predict the atomic

task based on the incorporated metrics and the component's analyzed sensitivity, as provided in Chapter 2.3. A t_w-second sliding window is applied to each sensor stream for each metric. The duration t_w required to segment the metrics is referred to as *window size*, while the stride duration t_s between each window (i.e., "the sliding action") is referred to as *step size*. The percentage of sensor data overlapping between two consecutive sliding windows can be determined using *step size* and *window size* (Equation 3.1).

$$overlap\ \% = \frac{t_w - t_s}{t_w} * 100 \tag{3.1}$$

The window and step sizes vary across the metrics depending on the metrics' sensitivity and sampling rate, as well as the individual algorithms. For example, inertial metrics are usually segmented into short-duration windows (i.e., $\leq 5s$), while the HRV metric typically requires at least thirty seconds of data in order to be sensitive to changes [86, 106, 107, 205, 258].

3.5.1 Cognitive Task Detection

The cognitive task recognition algorithm incorporates HRV, pupil dilations (left and right eyes), blink latency, and blink rate due to their correlation with mental workload and task difficulty [79, 86, 253]. Three time-based features: mean, standard deviation (std. dev.), and slope are extracted from the HRV, pupil dilations, and blink latency metrics. The mean and std. dev. capture the metrics' response to cognitive variations, while the slope captures the metrics' directional shift. A total of thirteen features, including the three time-based features extracted from HRV, pupil dilations, and blink latency metrics, as well as blink rate, are used to train a RF classifier to detect the cognitive tasks. A set of window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap are analyzed (see Chapter 4.2.1) [86, 106, 107, 205, 258]. Larger window sizes are used, because the HRV metric requires longer durations to be sensitive to changes.

3.5.2 Speech-Reliant Task Detection

The speech-reliant task detection algorithm incorporates the MFCCs metrics [256] and the three speech metrics (i.e., voice intensity, pitch, and speech rate) extracted from the Shure microphone headset (see Chapter 3.4). The microphone's speech audio is segmented into overlapping audio windows. Various window sizes (i.e., $t_w = \{1s, 3s, 5s, 10s, 15s\}$) with

a 50% overlap are investigated [61]. The algorithm employs a deep learning architecture (Figure 3.3) that consists of two parallel networks: i) a speech network and ii) a MFCC network. Incorporating speech patterns from the speech-based metrics in tandem with the audio stream can serve as key differentiators for detecting speech-reliant tasks and increase recognition accuracy (see Chapter 2.3.4.2) [6, 102].

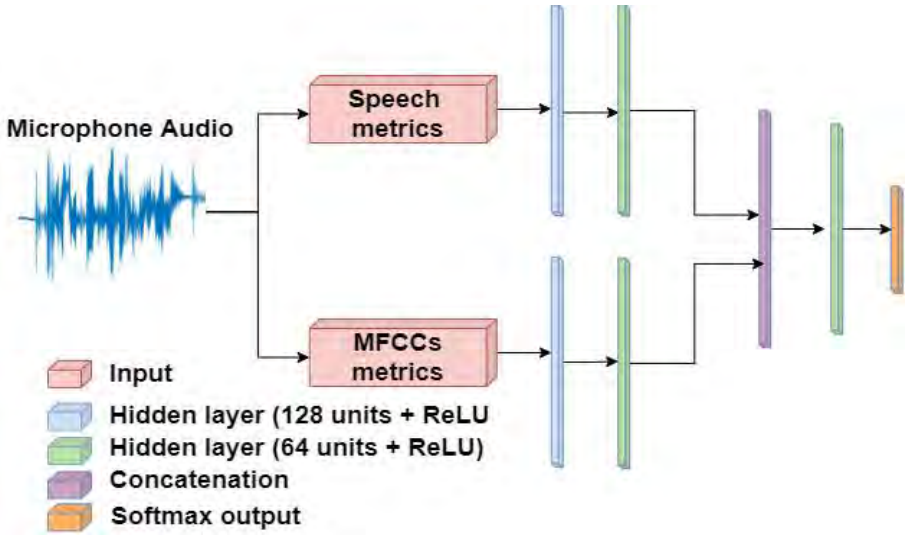


Figure 3.3: Speech-reliant task detection algorithm. The MFCCs and the speech-based metrics extracted from the microphone are fed into the MFCC and speech network, respectively. The output features from the networks are concatenated to detect the speech-reliant tasks.

The speech network's input layer consists of five neurons corresponding to the five features extracted from the segmented audio windows: voice intensity mean, voice intensity std. dev., pitch mean, pitch std. dev., and speech rate. The input layer is followed by two hidden layers with 128 and 64 neurons with Rectified Linear Units (ReLU) activation, respectively.

The MFCC network's input layer consists of forty neurons corresponding to the mean and std. dev. of the twenty MFCCs extracted per audio window. Two ReLU-activated hidden layers with 128 and 64 neurons, respectively, are added to the input layer. The output hidden layers from the speech and MFCC networks are concatenated and fed as input to another hidden layer with 64 neurons and ReLU activation is used to generate a high-level feature representation for the final output classification layer with softmax

activation. The ADAM optimizer [123] with a learning rate of 0.0005 is used for training the algorithm.

3.5.3 Auditory Task Detection

The auditory task detection algorithm incorporates the noise level metric obtained from the REED decibel meter and the log-Mel spectrogram metrics extracted from an ambient microphone (see Chapter 3.4). The algorithm employs a deep learning architecture (Figure 3.4) that consists of two parallel networks: i) a spectrogram network and ii) a noise network.

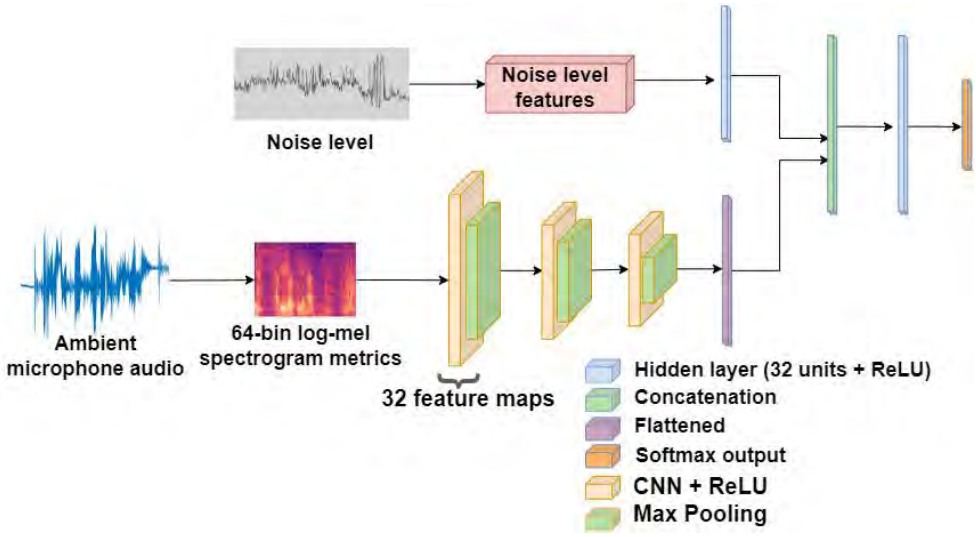


Figure 3.4: Auditory task detection algorithm. The log-scaled Mel spectrograms extracted from an ambient microphone are passed through three CNN layers, with 32 feature maps each. The CNN-generated convolutional features are flattened and concatenated with the noise level features and passed to a fully connected neural network to predict the tasks at the output layer.

The spectrogram network consists of three CNN layers with 32 feature maps each, as illustrated at the bottom of Figure 3.4. Each feature map is formed by convolving a 1×3 filter over each layer with ReLU activation [190]. Max pooling of size 1×2 is applied to reduce the feature representation at each layer, and a 50% dropout is applied after max pooling to avoid overfitting. The convolutional features from the CNNs are flattened into a 1-dimensional vector.

The noise network incorporates three time-based features (i.e., mean, std. dev., and the

slope) extracted from the noise level metric as input. The extracted input noise features are passed to a hidden layer with 32 ReLU-activated units, as depicted in Figure 3.4. The output hidden layers from the spectrogram and noise networks are concatenated and fed as input to another hidden layer with 32 ReLU-activated neurons. This hidden layer activation generates a high-level feature representation for the final output classification layer with softmax activation. The ADAM optimizer [123] with a learning rate of 0.0005 is used for training the algorithm. Various window sizes (i.e., $t_w = \{1s, 3s, 5s, 10s, 15s\}$) with a 50% overlap are investigated.

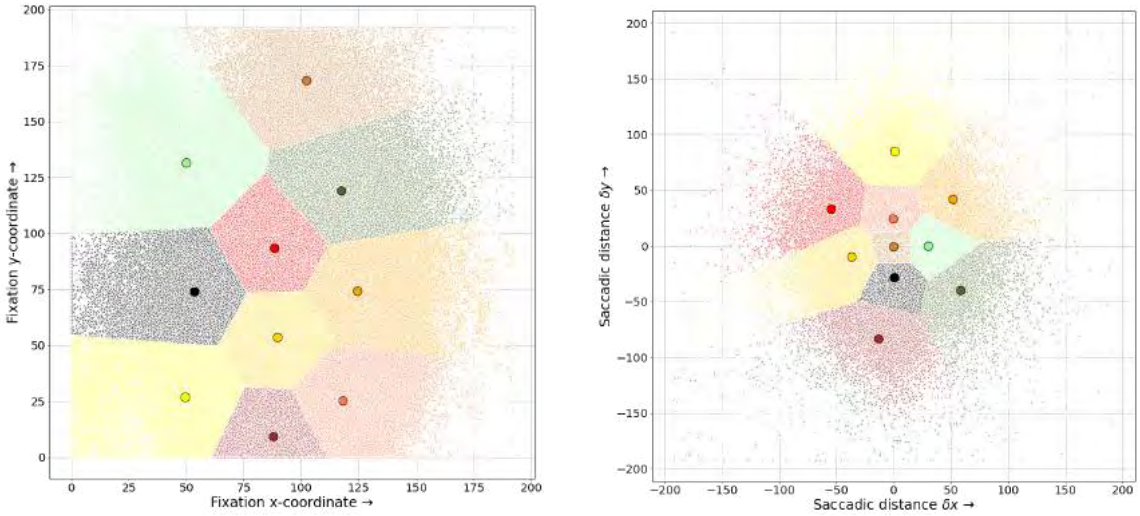
3.5.4 Visual task detection

The visual task recognition algorithm employs a multimodal approach, incorporating features extracted from the eye tracker's fixations, saccades, and the *Xsens*' forehead inertial metrics. The fixation and saccade gaze features capture the eye movements' spatio-temporal characteristics [253], while the inertial features provide additional context associated with the head movements [97, 138].

Initially, the participants' eye movements are analyzed by clustering the fixations and saccades separately using K-means clustering (N = 10 clusters resulted in the best classifier performance). The fixation f_x, f_y coordinates gathered across all participants are grouped into 10 clusters (Figure 3.5a), as were the saccades by grouping the saccadic distances (δ_x, δ_y) in the $\vec{x}$ and $\vec{y}$ axes (Figure 3.5b). Both clusters are used for constructing the fixation and saccade histograms during feature extraction.

Three different types of feature sets are extracted per sliding window: *fixation*, *saccadic*, and *inertial*. The fixation features are the fixation rate, fixation histogram, as well as mean, std. dev., and slope of the fixation duration and dispersion [30, 253]. The fixation dispersion is the angle (degrees) measured between a fixation's centroid and the two farthest points dispersed away from the centroid, while the fixation histogram is given by the frequency of the N = 10 fixation clusters. The saccadic features are the saccade length's mean, std. dev. and slope, as well as the saccadic histogram, which is given by the frequency of the N = 10 saccadic clusters. Finally, the inertial features consist of the accelerations' and angular velocities' mean, std. dev., and slope.

The extracted features are fed into a RF classifier with 100 decision trees, and a max depth of 500, where the parameters are chosen based on classifier performance. The window size has a significant effect on the number of fixations and saccades available for feature



(a) Fixation Cluster (b) Saccade Cluster

Figure 3.5: Eye movements are analyzed by clustering the fixations and saccades separately, using K-means clustering (N = 10).

extraction [253]. Smaller time windows allow for near real-time detection, but have poor accuracy, while longer windows have access to more information, resulting in better accuracy [30, 116, 120, 140, 174, 253]; thus, various window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap are investigated.

3.5.5 Gross Motor, Fine-Grained Motor, and Tactile Task Detection

Three wearable sensors are incorporated for detecting gross motor, fine-grained motor, and tactile tasks: i) the Xsens MTw Awinda motion capture system [207] that measures the IMU metrics using seventeen IMUs worn on various body parts, as depicted in Figure 3.2, ii) a Myo forearm sensor that measures sEMG and IMU data [239], and iii) the Biopac BioHarness™ that measures physiological metrics (i.e., heart rate, respiration rate, and postural magnitude). The signal dimension of the sliding windows varies across metrics and is given by: *Number of Channels* x (t_w * *sampling rate*), where *Number of Channels* represents the metrics sampled by the sensor, and *sampling rate* is the rate at which the metrics are obtained from the sensor. For example, the IMU's 3s window dimension is 6-channel IMU (i.e., three axes' acceleration and angular velocities) x (3s x 40 Hz) = 6 x

120. The windows are normalized by:

$$\hat{s_k^{I/E}} = \frac{s_k^{I/E} - \boldsymbol{\mu^{I/E}}}{\boldsymbol{\sigma^{I/E}}}, \tag{3.2}$$

where $s_k^{I/E}$ is either the 6-channel IMU, or the 8-channel sEMG sensor data window, $\boldsymbol{\mu^{I/E}}$ and $\boldsymbol{\sigma^{I/E}}$ are the IMU's or sEMG's mean and std. dev., respectively. The normalized data windows from the Xsens and Myo sensors serve as input to the algorithm. The algorithm employs a deep learning architecture that incorporates CNNs, where each network extracts features from the Xsens' and Myos' IMU and sEMG metrics.

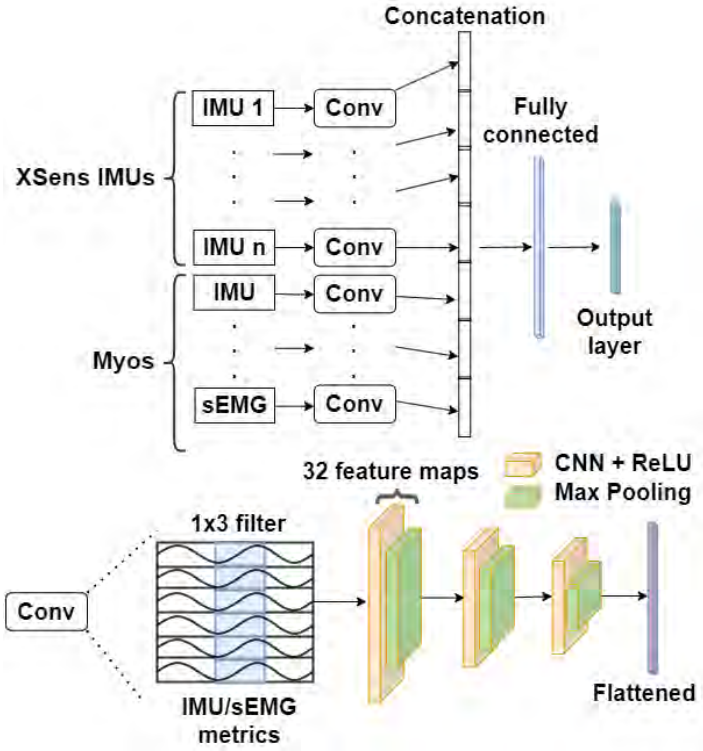


Figure 3.6: The task recognition algorithm, where CNNs extract features from the Xsens IMU trackers and Myos' forearm IMU and sEMG metrics. The three CNN layers have 32 feature maps each. The CNN-generated convolutional features are concatenated and passed to a fully-connected neural network to predict the tasks at the output layer.

The CNNs are three layers deep, with each layer consisting of 32 feature maps, as illustrated at the bottom of Figure 3.6. Each feature map is formed by convolving a 1 x

3 filter over each layer with ReLU as the activation function [190]. Max pooling of size 2 x 2 is applied to reduce the feature representation at each layer, and a 50% dropout is applied after max pooling to avoid overfitting. The convolutional outputs from the CNNs are flattened into a 1-dimensional vector in row-major order before being stacked (i.e., concatenated) to form a combined feature vector (indicated under the *Concatenation* operation in Figure 3.6). The concatenated feature vector is subsequently passed to a fully-connected neural network layer consisting of 32 neurons with ReLU activation. A 50% dropout is applied to the fully-connected layer to ensure the neurons do not rely on the features from any single network; thus, enabling the neurons to learn more robust features. Finally, the fully-connected layer is passed to an output softmax layer (see Figure 3.6) that computes the tasks' class scores. The number of tasks to be detected determined the number of neurons in the output layer. The deep learning model is trained to minimize cross-entropy loss. The ADAM optimizer [123] with a learning rate of 0.0005 is used to train the algorithm. Implementing separate CNNs permitted comparing the metrics' efficacy for detecting the tasks without altering the network dimensions.

The gross motor task detection algorithm combines the Bioharness' heart rate, respiration rate, and posture magnitude metrics with the upper and lower-body Xsens IMU data (i.e., waist, shoulders, thighs, calves, and feet). The Xsens has a 40 Hz sampling rate. Several window sizes (i.e., $t_w = \{1s, 2s, 3s, 5s, 10s\}$) with a 50% overlap (i.e., $t_s = 0.5 * t_w$) were analyzed [35, 43, 95]. The Bioharness' physiological metrics rely on time-based features (i.e., mean, std. dev., and slope) [192, 205]. The mean and std. dev. capture the metrics' response to the tasks, while the slope captures the metrics' directional shift. The Bioharness' low sampling rate (i.e., 1 Hz) and the analyzed shorter window sizes made it difficult for CNNs to extract meaningful features. For example, the physiological metrics' signal dimension for a 3s window is: 3 metrics x (3s x 1 Hz) = 3 x 3, which is too small to be convolved across the three CNN layers. Thus, the time-based features (mean, std. dev., and slope) extracted from the physiological metrics are combined with the Xsens IMU convolutional features at a later stage in the deep learning algorithm.

The fine-grained task recognition algorithm combines the Xsens IMU data from the wrists and hands of both arms, with the Myos' forearm IMU and 8-channel sEMG data. The Myos have a 100 Hz sampling rate. Five window sizes $t_w = \{1s, 2s, 3s, 5s, 10s\}$ with a 50% overlap were analyzed.

Four metrics are incorporated for tactile task detection: the left- and right-hand Xsens IMU, and the left and right forearm 8-channel Myo sEMG. The sensor data are segmented

into various window sizes ($t_w = \{0.5s, 1s, 1.5s, 2s, 3s\}$) with 50% overlap. Smaller window sizes were used due to the tactile tasks' shorter durations [62, 127, 130].

3.6 Fusion Algorithm

The individual component algorithms' predictions need to be fused in order to identify the set of all atomic tasks that are required as input for the *Composite and Concurrent* task detection algorithm. While the seven individual task detection algorithms are trained independently, assuming no interaction between component tasks, many of these tasks are interconnected. For instance, responding to a message over Walkie-Talkie involves the *auditory* task of listening to information, coupled with the *cognitive* task of processing the information. Similarly, picking up the Walkie-Talkie in the *fine-grained motor* component precedes the *tactile* task of pressing and holding it. Additionally, the individual task recognition algorithms employ different machine learning techniques that incorporate various metrics and sliding windows; therefore, the time to predict the atomic tasks (i.e., *prediction time*) differs across the algorithms. For example, the cognitive task recognition algorithm with a window size $t_w = 30s$ and 50% overlap provides atomic task detections every 15 seconds (i.e., $t_s = 0.5 * t_w$), while the tactile task recognition algorithm with a window size $t_w = 1s$ provides detections every half a second. Thus, a Graphical Neural Network (GNN) based *Fusion* algorithm was developed to leverage the relationships between components and accommodate time differences between individual algorithms, thereby consolidating task detections across all seven components.

GNNs are a class of deep learning models specifically designed to process and analyze structured data represented as graphs [240]. Unlike conventional neural networks that operate on grid-like data structures (e.g., images or sequences), GNNs handle complex relationships and dependencies between elements within a graph. A graph can be illustrated as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{v_1, v_2, ..., v_N\}$ is a set of N nodes, and $\mathbf{E} = \{e_1, e_2, ..., e_M\}$ is a set containing M edges. The nodes in a graph can represent entities (e.g., HRTs, robots, or tasks), while the edges capture relationships or interactions between these entities. An adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a sparse representation of a graph $\mathbf{G}$, where the adjacency matrix elements $\mathbf{A}_{i,j}$ denote the relationship (i.e., weight) between nodes v_i and v_j. The higher the weights, the stronger the relationship between the pairs of nodes.

GNNs typically operate on non-directed graphs; therefore, $\mathbf{A}$ is a symmetric matrix (i.e., $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$). GNNs leverage this graph structure to learn and reason about the data.

GNNs update the representation of each node iteratively by aggregating information from its neighboring nodes. A GNN's learning process involves two steps: i) message passing, and ii) node aggregation. During message passing, each node receives messages about the neighboring nodes' features via the edges. The messages are formed by convoluting a weight matrix $\mathbf{W}$ over neighboring nodes' features, a process known as graph convolutional filtering. The $\mathbf{W}$ is a learnable parameter optimized during the GNN training phase. The node aggregation step computes an updated feature representation for each node by summing or averaging the received messages at each node. Activation functions (e.g., ReLU and sigmoid) are applied to the updated feature representation to capture the non-linear relationship between the nodes. This process is repeated over multiple graph convolutional layers, which is analogous to the layers in a conventional neural network. The aggregation step allows nodes to capture and incorporate features from their local neighborhood, enabling them to learn rich representations that encode both local and global information.

Constructing the graph $\mathbf{G^f}$ required for the fusion algorithm to consolidate the atomic task detections is non-trivial. The graph $\mathbf{G^f}$ has seven nodes $\mathbf{V^f} = \{v_c, v_s, v_a, v_v, v_{gm}, v_{fm}, v_t\}$, where each node represents one of the seven activity components. The $v_c, v_s, v_a, v_v, v_{gm}, v_{fm}, v_t$ nodes correspond to the cognitive, speech, auditory, visual, gross motor, fine-grained motor, and tactile activity components, respectively. The corresponding adjacency matrix $\mathbf{A^f}$ is formed by taking the absolute values of Pearson's correlation coefficients between the activity components as:

$$\mathbf{A^f}_{i,j} = \left| \frac{\sum_{k=1}^{K}(w_i^k - \boldsymbol{\mu}_i)(w_j^k - \boldsymbol{\mu}_j)}{\sqrt{\sum_{k=1}^{K}(w_i^k - \boldsymbol{\mu}_i)^2 * \sum_{k=1}^{K}(w_j^k - \boldsymbol{\mu}_j)^2}} \right|, \tag{3.3}$$

where w_i^k and w_j^k are the respective workload values assigned to the i^{th} and j^{th} components when performing the k^{th} task. μ_i and μ_j are the i^{th} and j^{th} components' mean workload values across all tasks. The workload values indicate the task's difficulty level for each component. The tasks' workload values are determined a priori by a human task performance modeling tool (e.g., IMPRINT Pro [186]).

The GNN algorithm extracts features from the graph nodes using the graph convolutional filter operation repeatedly over multiple layers in order to predict the atomic tasks. A typical GNN contains L graph filtering layers with $L-1$ activation layers. The graph filtering and activation at the l^{th} layer is denoted as $h_l(\cdot)$ and $\alpha_l(\cdot)$, respectively. This learning process can be denoted as:

$$\begin{aligned} \boldsymbol{Z^{(l)}} &= h_l(\mathbf{A^f}, \boldsymbol{H^{(l-1)}}) \\ \boldsymbol{Z^{(l)}} &= \mathbf{A^f} * \boldsymbol{H^{(l-1)}} * \boldsymbol{W^{(l-1)}} \\ \boldsymbol{H^{(l)}} &= \alpha_l(\boldsymbol{Z^{(l)}}) \,, \end{aligned} \tag{3.4}$$

where $\mathbf{A^f} \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix (i.e., graph structure), $N = 7$ is the number of nodes in the graph $\mathbf{G^f}$, indicating the seven activity components. The operator $h(\cdot, \cdot)$ is the graph convolutional filter that takes the $\boldsymbol{H^{(l-1)}} \in \mathbb{R}^{N \times d_{l-1}}$ node feature and graph structure $\mathbf{A^f}$ as input, and outputs the new node feature $\boldsymbol{H^{(l)}} \in \mathbb{R}^{N \times d_l}$ at each layer. The parameter $\boldsymbol{W^{(l)}} \in \mathbb{R}^{d_l \times d_{l+1}}$ is the weight matrix learned at the l^{th} layer. $\boldsymbol{H^{(0)}}$ is the GNN fusion algorithm's input layer, where each node v_i in the $\mathbf{G^f}$ graph is described by a feature vector $\boldsymbol{H_{v_i}^{(0)}} \in \mathbb{R}^{d_{v_i}}$.

The GNN *Fusion* algorithm is formulated as a multi-label classification problem [263] that predicts multiple mutually non-exclusive labels (i.e., ≥ 0 atomic tasks may be present at any given instant). The GNN fusion algorithm's inputs $\boldsymbol{H_{v_i}^{(0)}}$ are the class probabilities from the individual algorithms' task detections. The class probabilities represent the associated classification confidence of the algorithms' detected tasks. The number of tasks detected by the individual task recognition algorithms varies across components (i.e., $\mathbb{R}^{d_{v_i}} \neq \mathbb{R}^{d_{v_j}}$); therefore, the class probabilities from the individual task recognition algorithms are projected into a 3-D latent space using a single hidden layer neural network, such that the class probabilities' dimensions across the seven components are equal (i.e., $\mathbb{R}^{d_{v_i}} = \mathbb{R}^{d_{v_j}} = \mathbb{R}^3$). The latent transformed class probabilities from the seven activity components are input to a three-layered GNN, whose graph structure is given by $\mathbf{A^f}$. The first layer $\boldsymbol{H^{(0)}} \in \mathbb{R}^{7 \times 3}$ consists of three node features per component, while the subsequent layers (i.e., $\boldsymbol{H^{(1)}}$ & $\boldsymbol{H^{(2)}}$) contain 32 node features each. All three layers are followed by a ReLU activation to model non-linearity. The final GNN layer $\boldsymbol{H^{(2)}}$ is collapsed (i.e., flattened) and passed to a fully connected layer with 32 neurons with ReLU activation to form the GNN feature vector. This feature vector is connected to seven softmax layers, *component output layers*, in order to predict the seven activity components' atomic tasks. The number of neurons at each *component output layer* will equal the number of atomic tasks detected by that respective component. The GNN fusion algorithm is trained end-to-end to minimize the joint cross-entropy loss. The ADAM optimizer [123] with a learning rate of 0.0005 is used to train the algorithm.

3.7 Composite and Concurrent Task Detection Algorithm

Composite and concurrent tasks typically consist of multiple atomic tasks occurring in parallel or sequentially over a period of time. Understanding such tasks requires capturing the temporal dependencies between the atomic tasks. A Temporal Convolutional Network [17, 148] based algorithm is proposed to detect the concurrent, composite tasks.

The *Fusion* algorithm predicts the atomic tasks as a vector-encoded list at regular time intervals across the seven components, which can be denoted by $\mathbf{x_t} = [x_{t_1}, x_{t_2}, ..., x_{t_K}]^\intercal$. x_{t_i} is the i^{th} component's atomic task at time t. K denotes the seven activity components. A series of all atomic tasks predicted over a time period T can be denoted as a matrix:

$$\mathbf{X} = < \mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_t}, ..., \mathbf{x_T} > , \tag{3.5}$$

where the T columns correspond to the time intervals, K rows correspond to the seven activity components, and each element x_{t_i} corresponds to the i^{th} component's atomic task at time t. Each time series $\mathbf{X}$ can be associated with one or more composite tasks. The *Composite and Concurrent* task detection algorithm's objective is to predict these composite tasks given the time series $\mathbf{X}$ as input. Formally, a sequence modeling network is any function

$$f : \mathcal{X}^\mathcal{T} \longmapsto \mathcal{Y}^\mathcal{T} \ni \hat{y_0}, \ldots, \hat{y_T} = f(x_0, \ldots, x_T)$$

that satisfies the causal constraint of y_t being dependent only on the current and previous inputs $x_0, \ldots, x_t$, and not on any future inputs $x_{t+1}, \ldots, x_{t+j}$. The sequence modeling network aims to learn the function f that minimizes the expected loss between the actual outputs and the predictions.

The Temporal Convolutional Network (TCN) is based on two convolutional principles: i) Causality, and ii) Dilation. *Causality* indicates that the output at any time step depends on the current and past inputs only, not future inputs, which is crucial for tasks like time series prediction where future data is unavailable. TCN employs 1-D *causal convolutions*, where output at time t is convolved only with elements from time t and earlier in the previous layer, with adequate zero padding to keep subsequent layers the same length as the previous layers [17]. A simple causal convolution cannot achieve a long effective history size without an extremely deep network or very large filters; therefore, the 1-D causal convolutions are *dilated* to enable an exponentially large receptive field [280]. Dilated

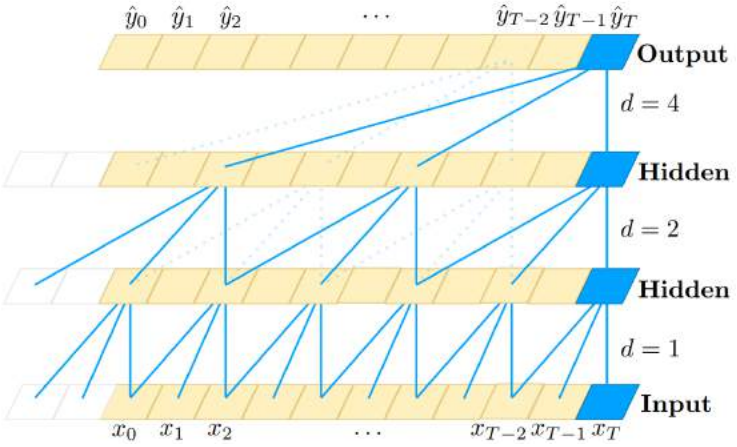


Figure 3.7: A dilated causal convolution with filter size $f = 3$, and dilation factors $d = 1, 2, 4$ increasing at each depth level. The effective sequence history at each layer is $(f - 1) \times d$, allowing the larger dilation at the top level to capture a wider range of inputs. The image is adapted from Bai et al.[17].

convolutions inflate the filter by inserting holes between the filter elements; thus, allowing the network to have a larger receptive field without increasing the number of parameters. A common strategy for TCN dilated convolutions is to increase the dilation rate d with the depth of the network (i.e., $d = 2^l$, where l is the network's depth), as depicted in Figure 3.7.

The TCN-based composite and concurrent task recognition algorithm, depicted in Figure 3.8, takes the time series $\mathbf{X}$ of size $K \times T$, where $K = 7$ is the number of activity components, and T is the overall window size that looks back over all the atomic tasks predicted across components over this period of time (see Equation 3.5). The encoder network converts the individual components' discrete atomic tasks into a continuous value by projecting them onto a 3-D latent space. The latent atomic values are passed through three TCN blocks with each block consisting of two 1-D dilated causal CNNs with ReLU activation. Each dilated causal CNN consists of 32 feature maps that are formed by convolving a filter of size f and dilation rate d, as shown in Figure 3.8. The dilation rate is increased exponentially at each level in order to expand the algorithm's receptive field. The TCN blocks also incorporate residual connections [83] to facilitate the flow of information through the network [280] in order to alleviate the vanishing gradient problem. The decoder network flattens the TCN blocks' output into a 32-D feature vector. Finally, the

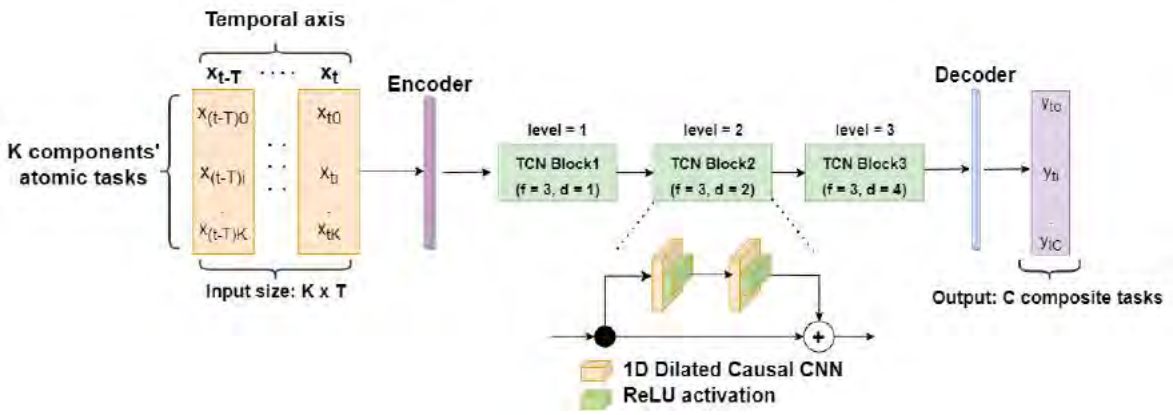


Figure 3.8: TCN composite and concurrent task recognition. The time series $\mathbf{X}$ of size K-components $\times$ T is passed as input to an encoder, which transforms the time series's discrete atomic tasks into a continuous value in latent space. The encoded value is passed 3 TCN blocks, each consisting of two 1-D dilated causal convolutions with ReLU activation and a residual connection. The filter size f is set to 3, but the dilation rate d is increased exponentially at each level. The TCN blocks' output is passed through a decoder network to predict the C composite tasks.

decoded feature vector is passed to an output sigmoid layer to predict multiple mutually non-exclusive C composite tasks (i.e., multi-label classification), as composite tasks can occur concurrently for a given series $\mathbf{X}$. The TCN algorithm is trained to minimize a weighted cross-entropy loss function. The ADAM optimizer [123] with a learning rate of 0.0005 is used to train the algorithm.

3.8 Summary

The developed individual task recognition and fusion algorithms incorporated reliable task recognition metrics obtained from wearable sensors to detect the atomic tasks from the contributing activity components. The algorithm is later extended to detect the concurrent, composite tasks using the TCN architecture. Two human-subjects evaluations, one supervisory-based and one peer-based, were conducted to assess the developed algorithms' capabilities. The experimental design and the algorithms' performance for the two evaluations are provided in Chapters 4 and 5.

Chapter 4: Supervisory-Based Experimental Analysis

A mixed-subjects supervisory human subjects evaluation was designed to manipulate participants' workload, based on the density of tasks. This evaluation design served two purposes, one to evaluate the ability to predict workload accurately (as a part of another effort) and to develop algorithms for detecting tasks accurately.

4.1 Experimental Design

The evaluation manipulated tasks, task density (i.e., workload), and the task density ordering as independent variables, see Table 4.1. The task environment is the NASA Multi-Attribute Task Battery-II (MATB-II) [46, 237], which simulates a supervisory-based human-machine team. The task density variable (i.e., *workload levels*) manipulated the number of tasks initiated during a specific time period. The workload was elicited by increasing and decreasing the NASA MATB-II tasks' frequency in three levels: i) Low or Underload (UL), ii) Medium or Normal load (NL), and iii) High or Overload (OL). The task density ordering (i.e., *workload ordering*) variable ensured that each task density (i.e., workload) transition (i.e., UL-NL, OL-UL) occurred exactly once. Participants completed a single 52.5-minute trial using an adapted NASA MATB-II version, where the trial consisted of seven consecutive 7.5-minute task density conditions. Three task density orderings were used:

- O_1: UL-NL-OL-UL-OL-NL-UL
- O_2: NL-OL-UL-OL-NL-UL-NL
- O_3: OL-UL-OL-NL-UL-NL-OL.

4.1.1 Task Environment

The supervisory task environment consisted of a modified version of the NASA MATB-II [46, 237], whose mission required a human operator to supervise a simulated remotely

Table 4.1: The independent variables for the supervisory-based evaluation

Type	Variable
within-subjects	Tasks Task density (i.e., workload)
between-subjects	Task density ordering

piloted aircraft. The NASA MATB-II mission consists of four composite tasks: tracking, system monitoring, resource management, and communication request. These composite tasks are composed of multiple atomic tasks and activity components. The original NASA MATB-II required participants to remain stationary, but real-life HRT scenarios require movement throughout the environment. The NASA MATB-II was modified to physically separate each NASA MATB-II task; thus, requiring participants to walk between two sets of tasks, *walking* task. This physical layout is depicted in Figure 4.1.

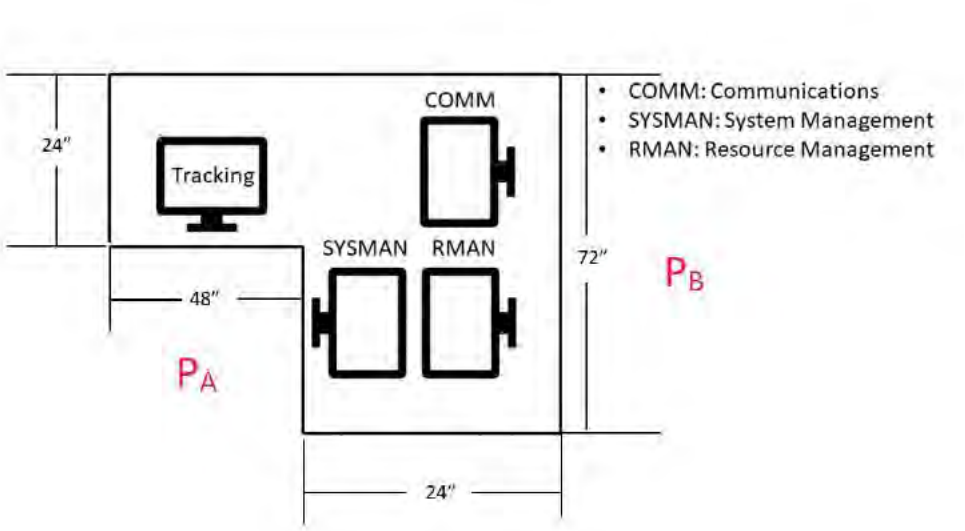


Figure 4.1: Physical Layout of the Modified NASA MATB-II. NOTE: P_A and P_B are the points between which participants walked back and forth to complete the tasks associated with the displays.

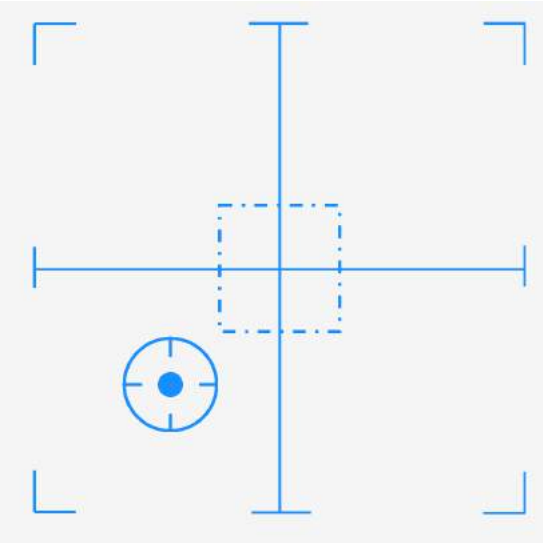
The modified version of the NASA MATB-II was coded using Python and PyGame in order to provide control over the task environment. Each NASA MATB-II task had a

computer monitor dedicated to it, where the computer monitors were stationed such that the participant was unable to visually see no more than two composite tasks simultaneously. This visual hindrance ensured that participants walked around the environment to complete the overall task objective. The required equipment (e.g., joystick or a keyboard) to complete each task was placed in front of the respective computer monitor. The table surfaces were approximately 4 feet from the floor. Participants were free to tilt the computer monitors up or down in order to accommodate height differences. The evaluation occurred in an empty conference room at an off-campus facility.

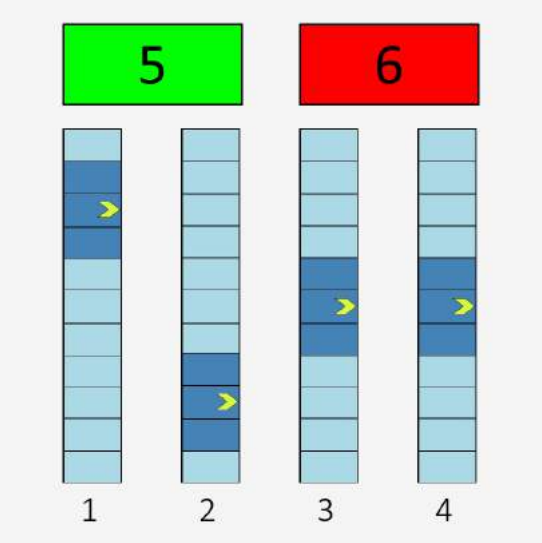
The tracking composite task, depicted in Figure 4.2a, required participants to keep the circle with a blue dot in the middle of the cross-hairs using a joystick and operated in two modes: *automatic* and *manual*. The *automatic* mode tracked the circle automatically without any participant input, while *manual* mode required the participant to track the circle physically using a joystick. The tracking composite task is composed of four atomic activity components: i) visual tracking, ii) cognitive association, iii) fine-grained joystick tracking, and iv) tactile joystick tracking. The UL condition, or low task density, required a total of 45s of manual tracking, with the remaining time for the condition being automated. The OL condition, high task density, had two 12s manual tracking sessions every minute, while the NL condition had one 20s session every minute, as determined using an Improved Performance Research Integration Tool (IMPRINT) Pro model [89].

The system monitoring composite task, shown in Figure 4.2b, required monitoring two colored lights and four gauges. If the green (L5) or the red light (L6) turned on, the value was out of range and required resetting. The four gauges had a randomly moving indicator, up and down, that typically remained in the middle. Participants reset a gauge if it was out of range (i.e., the indicator was too high or too low). The lights and gauges were reset by pressing the corresponding number key on the top row of the keyboard. The system monitoring task consists of four atomic tasks: i) visually inspecting the lights and gauges, ii) cognitive evaluation, iii) fine-grained keyboard usage, and iv) tactile keyboard stroke. The UL condition had only one out-of-range instance for the entire 7.5-minute session, OL had fifteen instances per minute, and NL had five instances per minute.

The resource management composite task included six fuel tanks (A-F) and eight fuel pumps (1-8), shown in Figure 4.2c. The arrow by the fuel pump's number indicated the direction fuel was pumped. Participants were to maintain the fuel levels of Tanks A and B by turning the fuel pumps on or off. Fuel Tanks C and D had finite fuel levels, while Tanks E and F had an infinite fuel supply. A pump turned red when it failed, during

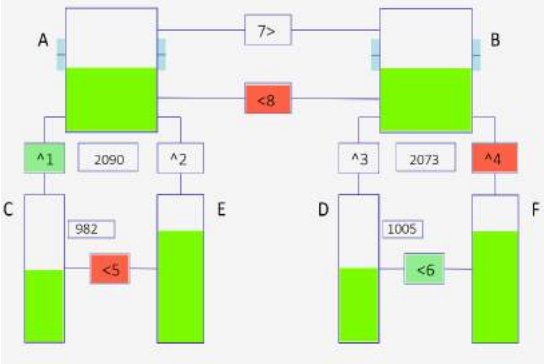


(a) Tracking

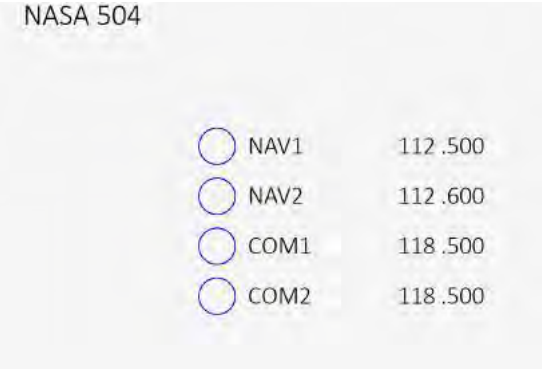


(b) System Monitoring

r



(c) Resource Management



(d) Communications

Figure 4.2: The NASA MATB-II Tasks

which it is unable to pump fuel. This composite task incorporates four atomic tasks: i) visual inspection, ii) cognitive evaluation, iii) fine-grained keyboard usage, and iv) tactile keyboard stroke. It is important to note that the fine-grained and tactile atomic tasks overlap between the system monitoring and resource management tasks, as the atomic tasks share similar motions and IMPRINT Pro model values; however, the number of atomic tasks involved and the order in which they appear differ. This composite task also operated in *automatic* and *manual* modes. The *automatic* mode maintained the tank levels by manipulating the pumps automatically without any participant input, while the *manual* mode required the participant to maintain the tank levels by toggling the pumps on or off by pressing the numbers (1-8) corresponding to the eight pumps using the number pad on the keyboard. The UL condition had 2 minutes of manual resource management with zero pumps failing, while the remaining time for the condition was automated. The OL condition had the resource management task on manual mode for the entire 7 min 30 seconds, with two or more pumps failing, while the NL condition had 3 min and 30 seconds of manual mode with at most two pumps failing every minute.

The communications composite task required listening to air traffic control requests for radio changes. The communication request was similar to: "NASA 504, please change your COM 1 radio to frequency 127.550." The original MATB communications task required no speech, but a required verbal response was added. An example response is: "This is NASA 504 tuning my COM 1 radio to frequency 127.550." Participants were to change the specified radio to the specified frequency by selecting the desired radio and using arrows to change the radio's frequency, as depicted in Figure 4.2d. Communications not directed to the participants' aircraft, as indicated by the call sign, were to be ignored.

The communication composite task can be decomposed further into two subtasks, communication request and communication response. The communication request subtask is composed of a single atomic task, the auditory communication request, while the communication response composite task is composed of five atomic tasks: i) visually locating the radio channels, ii) fine-grained mouse usage, iii) tactile mouse press, iv) a cognitive conversational element, and v) speech verbal response. The UL condition contained a total of one auditory communications request with one communication response task, the OL contained three auditory communications requests with at least two communication response tasks every minute, and the NL contained up to two auditory communications requests with only one communication response task per minute.

Finally, participants were required to walk around the tables to the other set of stations

(i.e., from P_A to P_B and vice-versa, as shown in Figure 4.1) whenever a *ping* sound occurred. Participants were free to move between the tasks at any time, but the *ping* sound enforced a mandatory transition to the other set of workstations. The walking task is a trivial case, containing only a gross motor component. The UL condition contained two walking requests, the OL condition incorporated seven walking requests per minute, and NL had two requests per minute.

Task timings and occurrences were chosen such that the correct workload condition, or task density, was elicited. The IMPRINT Pro tool was used to model the tasks for each workload level and ordering prior to conducting the evaluation. The IMPRINT Pro tool provided anchors to choose the correct *workload difficulty* value for a task. The anchor values are not normalized across components; thus, the association between a task and the workload value allocated varies significantly across the components. For example, a *conversation* is anchored to a speech workload value of 4.0, while *keyboard typing* is set to a fine-grained motor value of 7.0. The mapping between IMPRINT Pro's anchor values and tasks is provided in Table 4.2. The workload values for each NASA MATB-II task were chosen based on IMPRINT Pro's anchors. The chosen anchor values for the tasks by activity component are provided in Table 4.3.

The atomic tasks identified for each activity component are summarized in Table 4.4. The *Null* task associated with each activity component indicates an absence of the other atomic tasks. The detected tactile and fine-grained motor tasks may appear identical, but were not. The tactile interaction focused on a sense of touch, while the fine-grained motor movements involved the motion of wrists and hands for reaching and manipulating objects. For example, the *fine-grained motor* mouse use task involved grasping and manipulating the mouse, while the *tactile* mouse click task involved pressing the button. Essentially, the evaluated MATB-II task environment created a strong association between the activity components with some shared similar tasks, which may not be true in other domains.

4.1.2 Hypotheses

Five hypotheses were formed to evaluate the proposed individual task detection and fusion algorithms' ability to detect tasks correctly:

- H_1: Each individual task detection algorithm's accuracy will increase, as the window size increases, before reaching a point of diminishing returns.

Table 4.2: Mapping between Anchor values and Tasks

(a) **Gross Motor**

Value	Task description
1.0	Walking (even terrain)
2.0	Walking (uneven terrain)
3.0	Jogging (even terrain)
3.5	Heavy lifting
5.0	Jogging (uneven terrain)
6.0	Complex climbing

(b) **Fine-grained Motor**

Value	Task description
2.2	Discrete actuation (button)
2.6	Continuous adjustment (dial)
4.6	Tracking
5.5	Discrete adjustment
6.5	Writing
7.0	Keyboard typing

(c) **Tactile**

Value	Task description
1.0	Alerting
2.0	Simple discrimination
4.0	Complex symbolic information

(d) **Auditory**

Value	Task description
1.0	Detect sound
2.0	Orient to sound
3.0	Interpret speech (simple)
4.2	Verify audio feedback
6.0	Interpret speech (complex)
6.6	Discriminate sound
7.0	Interpret sound patterns

(e) **Visual**

Value	Task description
1.0	Register/Detect
3.0	Inspect/Check
4.0	Locate
4.4	Track
5.0	Read
6.0	Scan/Search monitor

(f) **Cognitive**

Value	Task description
1.0	Simple association
1.2	Alternative selection
3.0	Conversation
4.6	Evaluate (single aspect)
5.0	Rehearsal
6.0	Evaluate (multiple aspects)
7.0	Estimation/Calculation

(g) **Speech**

Value	Task description
2.0	Simple (1 -2 words)
4.0	Complex (sentence)

Table 4.3: IMPRINT Pro anchor values for the modified NASA MATB-II tasks.

Task	**Gross Motor**	**Fine-grained motor**	**Tactile**	**Visual**	**Cognitive**	**Auditory**	**Speech**
Tracking	0.0	4.6	2.0	4.4	1.2	0.0	0.0
System monitoring	0.0	2.2	2.0	3.0	4.6	0.0	0.0
Resource management	0.0	2.2	2.0	6.0	6.0	0.0	0.0
Communication	0.0	0.0	0.0	0.0	1.0	6.0	0.0
Communication response	0.0	2.6	2.0	4.0	3.0	0.0	4.0
Walking	1.0	0.0	0.0	0.0	0.0	1.0	0.0

Table 4.4: Atomic tasks identified for each activity component when using the modified NASA MATB-II task environment.

Activity Component	**Atomic tasks**
Gross motor	Walking, Null
Fine-grained motor	Joystick tracking, Keyboard usage, Mouse usage, Null
Tactile	Joystick tracking, Keyboard stroke, Mouse clicks, Null
Visual	Tracking, Inspect, Locate, Null
Cognitive	Association, Evaluation, Conversation, Null
Auditory	COMM request, Walk ping, Null
Speech	COMM verbal response, Null

- H_2: Each individual task detection algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the analyzed window sizes.
- H_3: The fusion algorithm's joint task optimization will improve the atomic task detection accuracy to $\geq 80\%$ across all seven components.
- H_4: The TCN-based composite and concurrent task recognition algorithm's overall accuracy will increase with the window size before reaching a point of diminishing returns.
- H_5: The TCN-based algorithm will detect composite tasks occurring concurrently with $\geq 80\%$ accuracy.

4.1.3 Metrics

The objective and subjective metrics were collected throughout the experiment. The objective metrics include the BioHarness' heart rate, HRV, respiration rate and posture magnitude, the Xsens' whole body inertial data (see Figure 3.2), the Myos' forearm inertial and sEMG data, the Pupil Core's pupil dilation, eye gaze, blink duration, and blink frequency, as well as the noise level and speech-based metrics (see Chapter 3.3). Accuracy was the primary dependent variable for assessing the algorithms' performance, while the confusion matrices compared the individual task recognition algorithms' accuracies and misclassifications by tasks.

The task recognition algorithms combined the incorporated metrics in $2^k - 1$ ways to analyze the metrics' impact, where k was the number of metrics incorporated. The gross motor algorithm incorporated four lower-body IMU metrics and the Bioharness' physiological features, which resulted in a total of *thirty-one* ($2^5 - 1 = 31$) gross motor metric combinations. The fine-grained motor algorithm incorporated the Xsens' hand and wrist IMUs, as well as the Myos' forearm sEMG and IMU, resulting in *fifteen* ($2^4 - 1 = 15$) metric combinations. These fifteen combinations were investigated across three handedness configurations: i) *left-only* that incorporated all four metrics from the left arm, ii) *right-only* that incorporated all four metrics from the right arm, and iii) *both* that incorporated all four metrics from both the arms, which resulted in a total of *forty-five* fine-grained motor metric data set combinations. Similarly, the tactile algorithm incorporated the Xsens' hand IMU and Myos' forearm metrics across three handness configurations,

resulting in *nine* combinations. The handedness analysis was not applicable to the other activity components. The visual and cognitive components incorporated three metrics each; thus, *seven* combinations each, while the speech component had two metrics, resulting in *three* combinations. The ambient audio was not recorded for this evaluation; therefore, the auditory component did not incorporate the spectrogram metric, so it only had *one* combination.

The subjective measure consisted of verbal in-situ workload ratings. The in-situ workload ratings required the participant to rate six demand channels (i.e., auditory, visual, speech, gross and fine-grained motor, tactile, and cognitive) from 1 (little to no demand) to 5 (extreme demand). The subjective metrics were not used in the experimental analyses.

The physically separated NASA MATB-II collected task performance metrics. The IMPRINT Pro model assumed all tasks were performed, even though actual participants may have missed one or more tasks, which can confound task labeling. The performance metrics reduce this confound and generate better ground truth data. The tracking task's performance was measured as the error in pixels between the center of the cross-hairs and the center of the object (Figure 4.2a). The system monitoring task's performance was determined by response time and failure rate. Response time was the number of seconds a participant took to click on a light or gauge, once the respective light or gauge went out of range. Failure rate represented the number of out of range lights and gauges that were not corrected. The resource management task's performance was determined by the amount of time (in seconds) fuel Tanks A and B were out of range (i.e., the fuel levels were not between 2,000 and 3,000 units). The number of failed communication requests (i.e., the participant failed to respond or the number of times the radio was tuned to the wrong frequency) determined the communications task performance.

4.1.4 Procedure

The participants completed a consent form and a demographic questionnaire upon arrival, after which participants were fitted with a BioPac Bioharness BT, Xsens Mtw Awinda motion trackers, a Pupil Core eye tracker, two Myo devices, and a Shure Microphone. A tutorial video described the NASA MATB-II tasks and how to accomplish the tasks. The tutorial video was followed by a 10-minute training session during which participants gained familiarity with the task environment, after which the 52.5-minute trial occurred. The training session cycled through the five tasks with each task occurring for one minute and

repeated the cycle one additional time. Participants completed a post-session questionnaire upon finishing the trial. In-situ workload ratings were verbally administered at 6 minutes into the trial and every 7.5 minutes after the initial rating.

4.1.5 Participants

Sixty-four participants (37 male, 24 female, and 3 non-binary) completed the experiment. The mean age was 29.80 (std. dev. = 10.24) with a range from 18 to 60. Thirty-four held a high school degree, fourteen held an undergraduate degree, fourteen held a master's degree, and five held a doctorate. Participants indicated the number of hours they use a desktop or laptop per week, as computer experience may impact task performance. The majority of participants (forty-five) indicated that they use computers for more than eight hours per week. Participants rated their video game skill level on average as 4.75 (std. dev. = 2.62) on a Likert scale (1-little to 9-expert). Thirty-four participants did not drink any caffeine the day of the experiment, while twenty-six participants drank at most 16 oz., and six participants drank more than 16 oz. Participants exercised on average 4.51 (std. dev. = 1.81) hours a week. Participants slept an average of 7.12 (std. dev.= 1.25) hours the night before the experiment and an average of 7.53 (std. dev. = 1.24) hours two nights prior. Participants rated current fatigue levels as 3.00 on average (std. dev. = 1.50) on a Likert scale from 1 (little to no) to 9 (extreme). Participants' dominant arm information was not gathered during the study; thus, the results are presented in terms of participants' left- vs. right-handed, rather than dominant vs. recessive arm. However, the general population is right-hand dominant for complex manual tasks [69, 215]; thus, 80-90% of the participants were assumed to be right-handed.

4.2 Results

The task recognition algorithms are validated using the *leave-one-subject-out* cross-validation scheme, where the average accuracy is reported by training the algorithm repeatedly on all, but one participant's data and validating using the left-out participant's data [81]. The confusion matrices compare the individual task recognition algorithms' accuracies and misclassifications by tasks. A Friedman's analysis of variance by ranks test is used to determine statistical significance in accuracies between results. Significant results were further analyzed using the Wilcoxon signed-rank test to identify the specific significant dif-

ferences. The non-parametric statistical tests ensured that the outcomes were unaffected by the accuracy distribution across participants. Cohen's d measured the effect sizes.

4.2.1 Cognitive Task Recognition

The cognitive task recognition algorithm incorporated thirteen features extracted from HRV, pupil dilations (left and right eyes), and blink metrics (see Chapter 3.5.1). The features were fed into a RF classifier that was trained to predict one of the four cognitive tasks: i) Association, ii) Conversation, iii) Evaluation, and iv) Null (described in Chapter 4.1.1) for each window. The evaluated window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance.

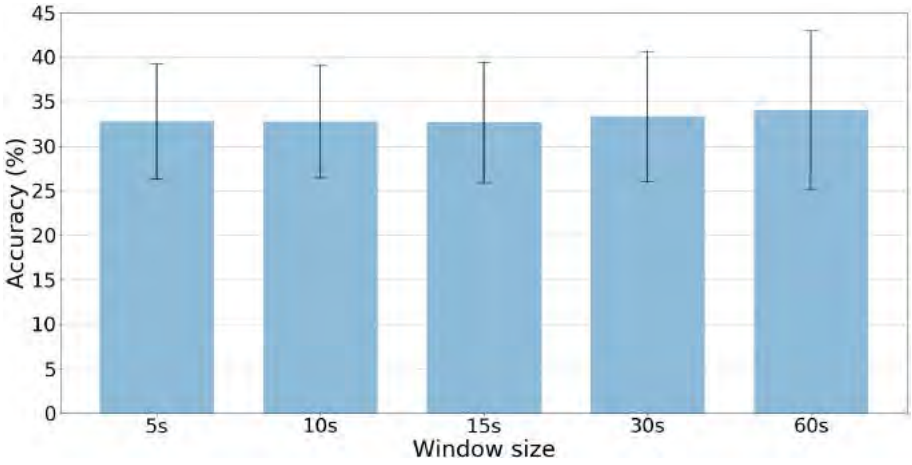


Figure 4.3: Cognitive task recognition accuracy % (mean (std. dev.)) by window size.

The RF algorithm's accuracy decreased slightly from 5s (32.80%) to 15s (32.68%), before increasing and achieving its peak accuracy at the 60s window size (34.09%), as shown in Figure 4.3. The Friedman's test indicated that there was no significant difference between window sizes ($\chi^2(4, 60) = 4.32, p = 0.36$).

The RF algorithm's confusion matrices for the evaluated window sizes were analyzed to identify the best-performing window size (see Figure 4.4). None of the evaluated window sizes detected the cognitive tasks reliably. Most window sizes classified three out of the four tasks with approximately 30% accuracy. All five window sizes had higher misclassification rates for the Evaluation task, which was confused with the Association and Null tasks. Among the five window sizes, the 15s window size variant had the lowest confusion rate.

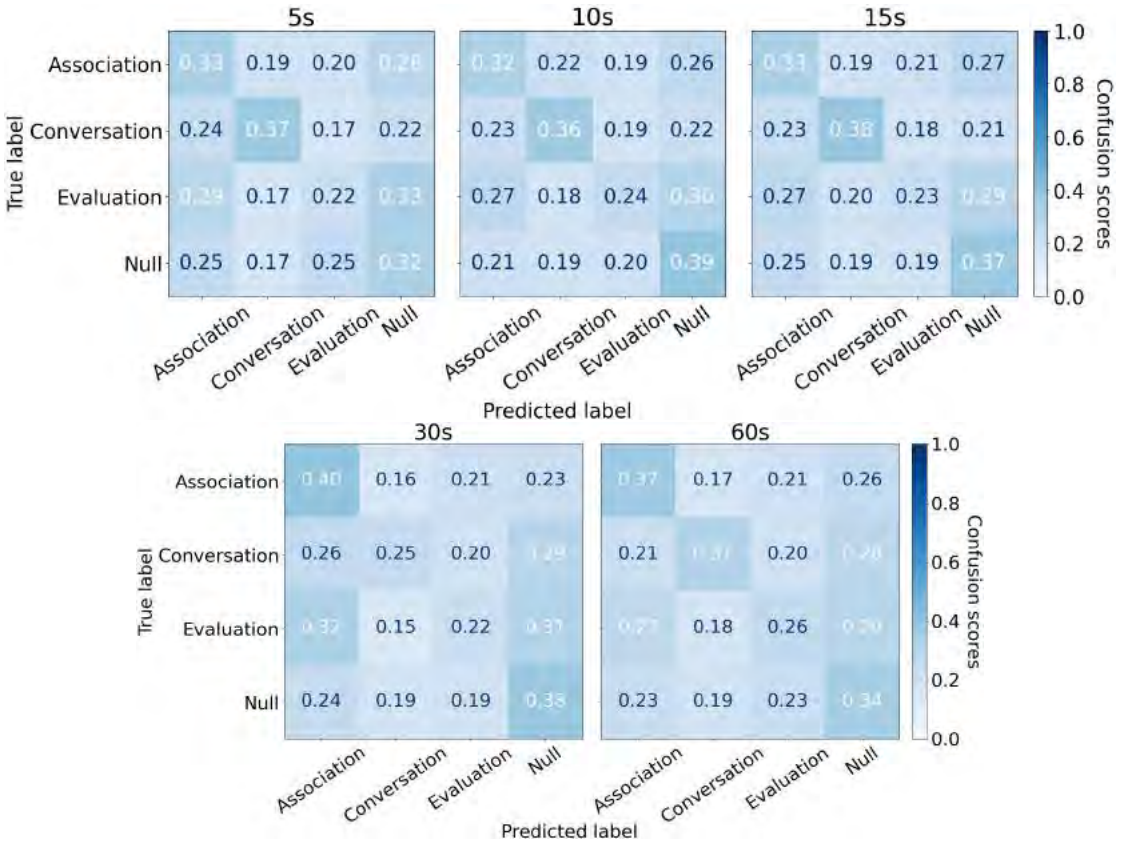


Figure 4.4: The cognitive task recognition confusion matrices when HRV, pupil dilation, and blink rate metrics are incorporated for the evaluated window sizes.

Table 4.5: Cognitive task recognition accuracy (mean % (std. dev.)) by the incorporated metrics using 15s window size RF algorithm aggregated across participants. The highest accuracy is highlighted in Bold.

Metrics	Accuracy
HRV	26.52 (4.41)
Pupil dilation	**31.11 (7.38)**
Blink	28.74 (5.20)
HRV + Pupil dilation	31.44 (7.07)
HRV + Blink	29.46 (6.60)
Pupil dilation + Blink	**32.27 (7.91)**
HRV + Pupil dilation + Blink	**32.68 (6.74)**

The incorporated metrics can impact the RF algorithm's performance. Using the 15s window size, the RF algorithm was trained by combining the metrics in several combinations. A total of seven combinations were evaluated by incorporating the metrics individually and by combining two and three metrics at a time (see Table 4.5).

The highest individual metric accuracy (31.11%) was achieved by the pupil dilation metrics, while the HRV metric had the lowest accuracy (26.52%). The Wilcoxon signed-rank test revealed that the pupil dilation metrics' accuracy was significantly higher than the HRV ($p < 0.01$) and blink ($p = 0.02$) metrics, while the HRV metrics' accuracy was significantly lower than the pupil dilation ($p < 0.01$) and blink ($p = 0.04$) metrics' accuracies.

The highest accuracy (32.27%) when incorporating two metrics was achieved when the pupil dilation and blink metrics, while the lowest accuracy (29.46%) was recorded when the HRV and blink metrics were combined. The Wilcoxon signed-rank test revealed that the accuracy when combining the pupil dilation and blink metrics was significantly higher ($p = 0.03$) than the HRV and blink metrics combination, but was not significant otherwise. The test also revealed that the pupil dilation and blink combination's accuracy and the accuracy of all three metrics combined did not differ significantly. Generally, adding additional metrics to pupil dilation did not affect the algorithm's accuracy significantly.

4.2.1.1 Discussion

Hypothesis $\mathbf{H_1^C}$ predicted that the RF algorithm's accuracy will increase, as the window size increases before reaching a point of diminishing returns, which was not supported. The algorithm's poor classification performance can be attributed to the selected metrics' (i.e., HRV, pupil dilation, blink) features that may not be suitable for cognitive task recognition, especially if tasks change frequently.

Hypothesis $\mathbf{H_2^C}$ predicted that the RF algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The algorithm's accuracy when predicting the tasks, regardless of the window size, was only 5% to 10% better than randomly guessing the tasks. Labeling cognitive tasks is non-trivial and highly uncertain, as it is difficult to determine when exactly the human began the mental processes prior to executing a task. This uncertainty in cognitive task labeling may have also exacerbated the poor performance.

The performance analysis by metrics indicated that pupil dilation was the most useful

metric and incorporating additional metrics did not improve the algorithm's performance. The incorporated metrics may not be responsive enough to identify cognitive changes within a short duration (i.e., *reactive*), such that the tasks can be detected before task switching. Assuming the tasks occur for a prolonged period of time is a poor assumption for real-time, dynamic, and uncertain domains; therefore, other metrics that conform with the reactivity criterion in addition to all the other criteria must be investigated in order to develop more accurate cognitive task recognition. An alternative is to detect cognitive tasks indirectly based on the other atomic component task detections using the GNN-based fusion algorithm (see Chapter 4.2.8).

4.2.2 Speech Task Recognition

The speech-reliant task detection algorithm incorporated the MFCCs' mean and std. dev. and the five features extracted from the speech-based metrics. The features were fed into a deep learning algorithm to predict two tasks: i) COMM response and ii) Null (described in Chapter 4.1.1). Five window sizes ($t_w = \{1s, 3s, 5s, 10s, 15s\}$) with a 50% were used to evaluate the impact of the window size on algorithm's performance; however, the 15s window size had no instances for the COMM response task, as participants spoke for $< 15s$ for all COMM requests received. Therefore, the 15s window is excluded from the analysis. It must also be noted that the evaluated domain contained simple speech ($< 1second$) in the in-situ subjective ratings given by the participants (see Chapter 4.1.1). However, the in-situ ratings were not included in this analysis, due to the lack of data annotation.

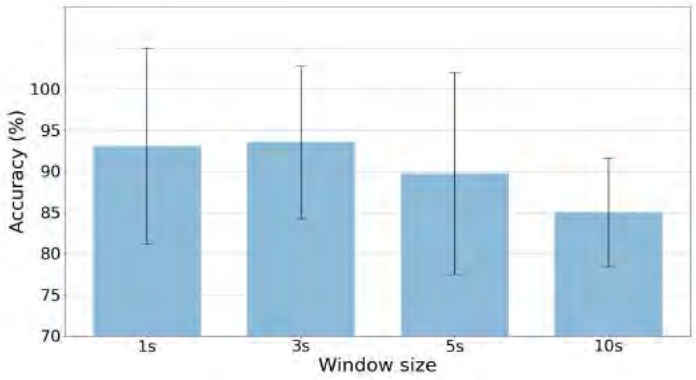


Figure 4.5: Speech-reliant task recognition accuracy by window size when incorporating the speech-based and MFCC metrics.

The average time taken to verbally respond to the COMM request was 3.07 (std. dev. = 1.52) seconds, with the shortest and longest COMM response being 1.01 and 8.15 seconds, respectively. The algorithm's accuracy increased with window size until the 3s window and decreased for the 5s and 10s window sizes. The Friedman's test indicated that the accuracies varied significantly between window sizes ($\chi^2(3, 32) = 19.56, p < 0.01$). The Wilcoxon signed-rank test indicated that the 10s window size's accuracy was significantly lower than the others with a large effect size ($p < 0.01$, $1.90 <$ Cohen's $d < 2.44$), while the 1s and 3s window sizes' accuracies were significantly higher than the 5s and 10s window sizes with a medium to large effect size ($p < 0.01$, $0.31 <$ Cohen's $d < 2.44$). No other differences were significant. The confusion matrices between the 1s, 3s, 5s, and 10s window sizes (see Figure 4.6) also concur with the statistical analysis in that the 1s, 3s, and 5s window sizes distinguished the COMM response and Null tasks with $\geq 90\%$ accuracy, with the 1s and 3s outperforming the rest.

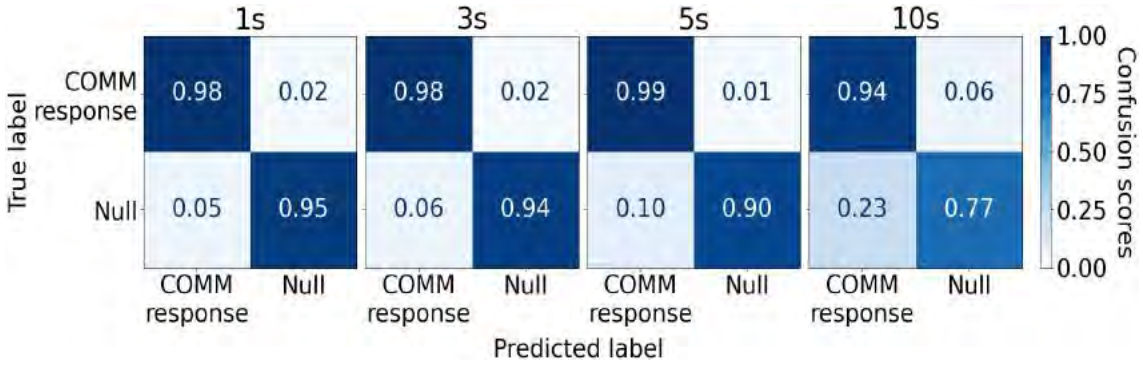


Figure 4.6: The speech task recognition confusion matrices for the 1s, 3s, 5s, and 10s window sizes.

Table 4.6: Speech-reliant task recognition accuracy (mean % (std. dev.)) by the incorporated metrics for the 3s window aggregated across participants. The highest accuracy is highlighted in Bold.

Metrics	Accuracy
Speech-based	**94.57 (2.07)**
MFCCs	94.08 (6.40)
Speech-based + MFCCs	93.57 (9.25)

Using the 3s window size, the algorithm was trained by incorporating the metrics

individually and by combining them (see Table 4.6). The algorithm trained only on the speech-based metrics achieved the highest accuracy (94.57%), while combining both speech-based and MFCCs metrics had the lowest accuracy (93.57%); however, the Friedman's test revealed that the metrics' accuracies did not differ significantly.

4.2.2.1 Discussion

Hypothesis $\mathbf{H_1^S}$ predicted that the algorithm's accuracy will increase, as the window size increases before reaching a point of diminishing returns. This hypothesis was supported. The 1s and 3s window sizes appear to be the optimal window size for speech-reliant task recognition. The average COMM response length was calculated to be ∼3 seconds; therefore, the 3s window is recommended for this domain. However, the in-situ ratings were not included in this analysis, which may alter the results. The in-situ ratings will be analyzed for a peer-based evaluation in order to determine the optimal window size that can be used for detecting both simple and complex speech tasks.

Hypothesis $\mathbf{H_2^S}$ predicted that the algorithm will detect events with $\geq 80\%$ classification accuracy for at least one of the window sizes. The hypothesis was fully supported, as the algorithm detected the tasks with high sensitivity ($> 80\%$) regardless of the window size. The algorithm's high accuracy can be attributed to the NASA MATB-II task environment's limited number of speech tasks and the COMM response was fairly standard complex speech that did not vary across participants, as it may in other tasks.

The complex speech required participants to utter almost an entire sentence to confirm the radio frequency change, which may not be the case for real-world scenarios, where human teammates may communicate in cryptic phrases with fewer words. The peer-based evaluation included additional simple (speech with fewer words) and complex tasks with varying complexities, lengths, tones, and syllables to better assess the algorithm's viability and the metrics' impact on the intended domain (see Chapter 5.2.2).

4.2.3 Auditory Task Recognition

The spectrogram metrics, described in Chapter 3.5.3, were not employed for this analysis, as the ambient audio was not gathered for the supervisory evaluation. A temporary auditory task recognition algorithm was developed for this evaluation. The algorithm incorporated three time-based features (i.e., mean, std. dev., and slope) extracted from the noise level

metric obtained via the decibel meter. The features were fed into a RF classifier [256] trained to predict one of the three auditory tasks: i) COMM request, ii) Ping (played to trigger the *Walking* task), and iii) Null (described in Chapter 4.1.1) for each window. The classifier with 100 decision trees and a max depth of 500 performed the best. The evaluated window sizes $t_w = \{1s, 3s, 5s, 10s, 15s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance.

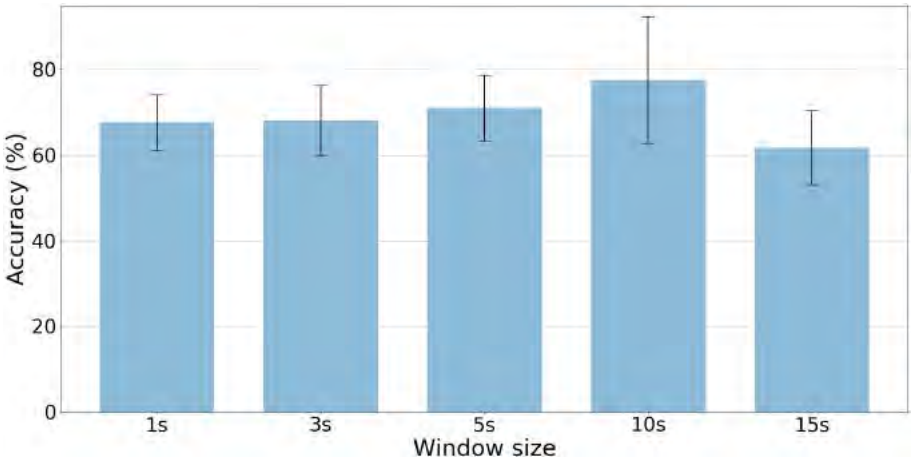


Figure 4.7: Auditory task recognition accuracy % (mean (std. dev.)) by window size.

The RF algorithm's accuracy increased gradually until the 10s window size (77.56%) and decreased at the 15s window size (61.81%), as depicted in Figure 4.7. The Friedman's test indicated a significant difference in accuracies between window sizes ($\chi^2(4, 60) = 15.27, p < 0.01$). The Wilcoxon signed-rank test indicated that the 10s window size's accuracy was significantly higher than all the other window sizes with a medium to large effect size within the RF ($p < 0.01$, $0.55 <$ Cohen's $d < 0.98$), while the 5s window size's accuracy was significantly higher than the 1s and 3s window sizes with a small effect size ($p < 0.01$, $0.36 <$ Cohen's $d < 0.47$). No other differences were significant.

The RF algorithm's confusion matrices for the 3s, 5s, and 10s window sizes were analyzed to identify the best-performing window size (see Figure 4.8). The 1s and 15s window sizes' confusion matrices are provided in Appendix A Figure A.1. The confusion matrices indicated that the algorithm had high classification rates for the *COMM* and *Null* tasks, while the *Ping* task was consistently confused with the *COMM* task across all window sizes. Overall, the RF's 10s window size had a better classification rate by task.

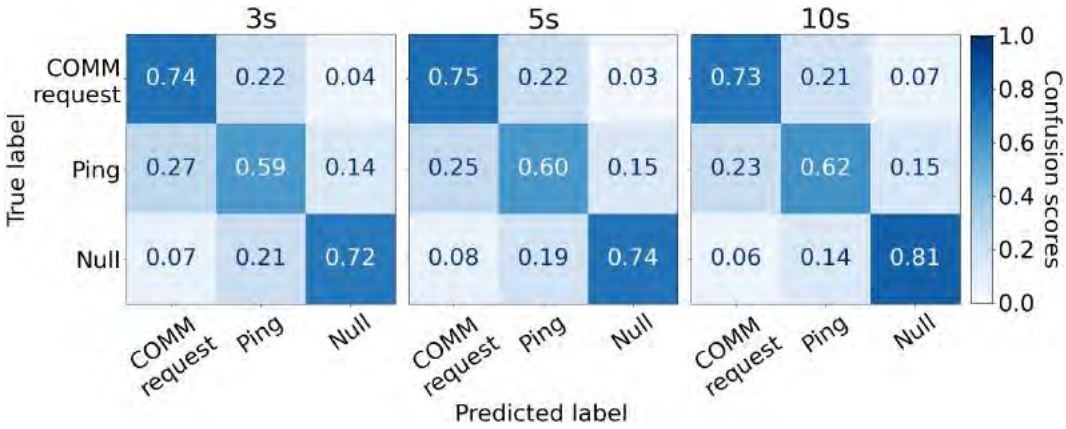


Figure 4.8: The RF auditory task recognition confusion matrices for the 3s, 5s, and 10s window sizes.

4.2.3.1 Discussion

Hypothesis $\mathbf{H_1^A}$ predicted that the RF algorithm's accuracy will increase, as the window size increases before reaching a point of diminishing returns, which was partially supported. The 10s window size variant had the best performance, overall and by tasks; thus, it is the window size for a dynamic task environment when the noise level is the only metric incorporated by the algorithm to detect the auditory tasks. However, based on the literature reviewed, the smaller window sizes $\leq 5s$ are hypothesized to perform better when the algorithm incorporates spectrogram metrics [82, 144, 159], as it can identify auditory changes within a short duration, so that a task can be detected before it gets switched.

Hypothesis $\mathbf{H_2^A}$ predicted that the RF algorithm will detect events with $\geq 80\%$ classification accuracy for at least one of the analyzed window sizes, which was not supported. The less-than-ideal accuracy (i.e., 2% lower than the expected accuracy) can be attributed to the noise level's ability to distinguish the auditory tasks purely based on loudness (i.e., amplitude). The noise level metric fails to capture other characteristic features (e.g., event's frequency and spectral envelope) that are typically used for detecting auditory tasks [82, 91, 144, 159]. Further, the evaluated domain entailed the experimenter interrupting the participants for in-situ subjective ratings (see Chapter 4.1.1), which is an auditory task. However, the in-situ auditory interrupts were not included in this analysis, which may have skewed the results.

Combining the spectrogram metrics from an ambient microphone with the noise level

metric can potentially increase the auditory task detection accuracy by providing the algorithm with the characteristic ambient sounds and their associated loudness present in the task environment. This analysis did not evaluate the actual auditory task recognition algorithm (Chapter 3.5.3) due to the lack of ambient microphone audio; however, the peer-based evaluation (see Chapter 5.2.3) incorporated the spectrogram metrics to detect auditory tasks with varying lengths and noise characteristics.

4.2.4 Visual Task Recognition

The visual task recognition algorithm incorporated features extracted from the eye tracker's fixation and saccade metrics, as well as the Xsens' head motion tracker's inertial metrics. The features were fed into a RF classifier that was trained to predict one of the four visual tasks: i) Tracking, ii) Inspect, iii) Locate, and iv) Null (described in Chapter 4.1.1) for each window. The evaluated window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance.

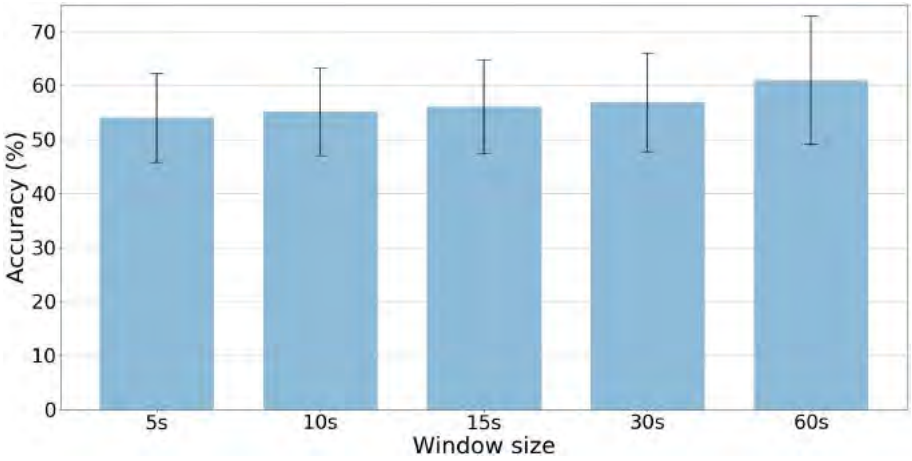


Figure 4.9: Visual task recognition accuracy % (mean (std. dev.)) by window size.

The RF algorithm's accuracy increased gradually with window size, achieving the highest accuracy (61.01%) at the 60s window size (see Figure 4.9). The Friedman's test revealed that the accuracies were significantly different between window sizes ($\chi^2(4, 60) = 21.47, p < 0.01$). The Wilcoxon's test indicated that the 5s window size's accuracy was significantly lower than all other window sizes with a small to medium effect size ($p < 0.01$, $0.14 <$ Cohen's $d < 0.68$), while the 60s window size's accuracy was significantly higher than all

other window sizes with a medium effect size ($p < 0.01$, $0.38 <$ Cohen's $d < 0.68$). No other differences were significant.

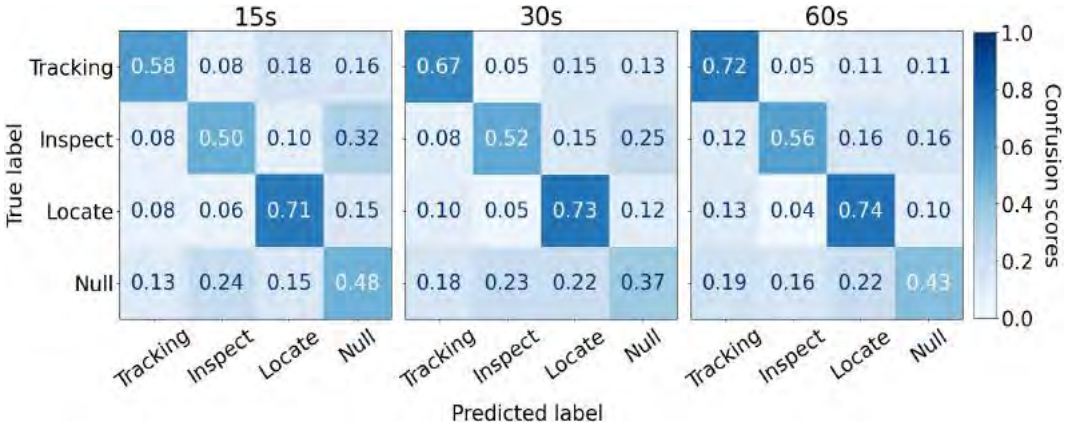


Figure 4.10: The visual task recognition confusion matrices when fixation, saccades, and inertial metrics are incorporated for 15s, 30s, and 60s window sizes.

The confusion matrices (see Figure 4.10) indicated that the 60s window size variant had the least confusion rate across most tasks, resulting in up to 5 - 12% increase in the tasks' accuracies when compared to the 15s and 30s window sizes. All other tasks had similar confusions and accuracies. Therefore, the 60s window size's recognition rate by task was better than the other window sizes. The 5s and 10s window sizes had subpar accuracies by tasks (see Appendix A Figure A.2).

Table 4.7: Visual task recognition accuracy (mean % (std. dev.)) by the incorporated metrics for the 60s window RF algorithm aggregated across participants. The highest accuracy is highlighted in Bold.

Metrics	Accuracy
Fixation	39.82 (12.1)
Saccades	48.49 (14.14)
Inertial	**53.80 (10.35)**
Fixation + Saccades	48.49 (13.51)
Fixation + Inertial	56.14 (10.33)
Saccades + Inertial	**60.18 (11.38)**
Fixation + Saccades + Inertial	**61.01 (11.86)**

Using the 60s window size, the RF algorithm was trained by combining the metrics in several combinations. A total of seven combinations were evaluated by incorporating the

metrics individually, and by two and three metrics simultaneously as shown in Table 4.7.

The analysis by individual metric found the highest accuracy (53.80%) was attained by the head inertial metrics, while the fixation metrics had the lowest accuracy (39.82%). The Wilcoxon signed-rank test revealed that the head inertial metrics' accuracy was significantly higher ($p < 0.01$) than the other two metrics, and the saccade metrics' accuracy was significantly higher ($p < 0.01$) than the fixation metrics.

The highest accuracy (60.18%) when incorporating two metrics simultaneously was achieved by combining the saccades and head inertial metrics, while the lowest accuracy (48.49%) was recorded when the fixation and saccades were combined. The Wilcoxon signed-rank test revealed that the saccades and head inertial combination's accuracy was significantly higher ($p < 0.01$), than the remaining two combinations. The test also revealed that the saccade and head inertial combination's accuracy and the accuracy of all three metrics combined did not differ significantly.

4.2.4.1 Discussion

Hypothesis $\mathbf{H_1^V}$ predicted that the RF algorithm's accuracy will increase, as the window size increases, before reaching a point of diminishing returns. The hypothesis was supported, as the accuracy continued to increase until the 60s window size. The RF algorithm's 60s window size had the best overall performance; thus, it is the recommended window size using the current metrics for the evaluated supervisory domain.

Hypothesis $\mathbf{H_2^V}$ predicted that the RF's algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes. This hypothesis was not supported, as the RF algorithm's maximum accuracy was only $\sim$60%, regardless of the window size. The algorithm's poor performance can be attributed to two factors. First, participants' eye and head movement patterns may not have been distinct enough between tasks, indicating that the multi-tasking nature may have had a negative impact on detection accuracy. Second, labeling the visual tasks is non-trivial and highly uncertain, as it is difficult to determine when exactly the participant's visual processes began prior to task execution. This labeling uncertainty may have also exacerbated the poor performance.

It is important to determine the incorporated metrics' ability to detect the tasks reliably. The selected metrics were inadequate to capture the participants' visual behavior in a multi-tasking environment. The per-metric analysis indicated that Xsens' head motion inertial data was the most useful, followed by the saccade and fixation metrics. The visual task

detection analysis determined that the incorporated metrics are less responsive for reliably detecting tasks in a dynamic, multi-tasking environment (i.e., switching tasks frequently).

Identifying the appropriate window size for each algorithm informs how the metrics must be segmented, such that the features extracted are representative of the tasks being detected. The analysis indicated that the incorporated metrics generally require larger window sizes ($> 30s$) to assimilate the context needed to detect the visual tasks reliably. The evaluation data is highly uncertain due to the rapid task switching and accompanied labeling difficulty; therefore, it was harder for the algorithm to assimilate the required context at lower window sizes. The 60s window size is recommended, because it is large enough to provide the algorithm with the required context, amidst the uncertainty.

Visual tasks will have different durations. A short task (e.g., inspection) may require a smaller window, so that the task is not overshadowed (e.g., confused) by all the unrelated data; therefore, it may be necessary for the task recognition algorithm to use an adaptive sliding window method [170, 195]. An adaptive sliding window will permit for expanding and contracting of the window size, based on the task, which may lead to more accurate detection.

4.2.5 Gross Motor Task Recognition

The gross motor task recognition algorithm incorporated the Xsens' pelvis, thighs, calves, and feet IMU metrics, along with the Bioharness' physiological metrics. The algorithm predicted two gross motor tasks: i) Walking and ii) Null. Window sizes, $t_w = \{1s, 2s, 3s, 5s, 10s\}$, with a 50% overlap, were investigated for analyzing the window size's impact on the algorithm's performance.

Overall, the algorithm's accuracy increased until the 3s window size (80.97%) and decreased to 77.49% for the 10s window size when incorporating the physiological and all four lower-body IMU metrics (see Figure 4.11). The Friedman's test indicated a significant accuracy difference between window sizes ($\chi^2(4, 60) = 33.10$, $p < 0.01$). The Wilcoxon signed-rank test found that the 2s and 3s window size's accuracies were significantly higher than the 1s and 10s window sizes ($p < 0.01$, $0.23 <$ Cohen's $d < 0.39$), and the 5s window size's accuracy was significantly higher than the 10s window size ($p < 0.01$, Cohen's $d = 0.31$). No other differences were significant.

The confusion matrices (see Figure 4.12) indicated that the 2s, 3s, and 5s window sizes had similar task-wise accuracies. However, the 3s window size performed the best in 30

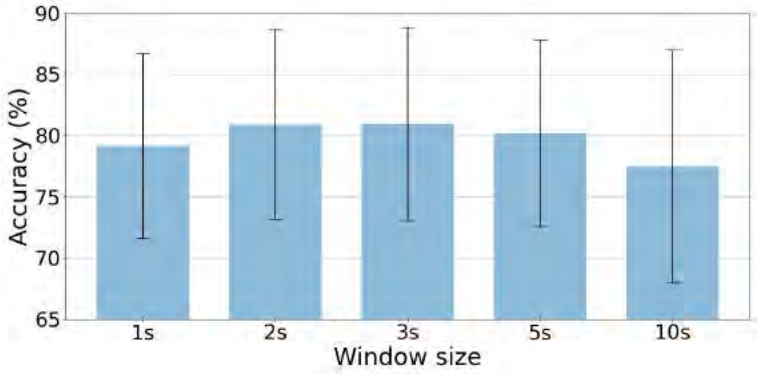


Figure 4.11: Gross motor task recognition accuracy by window size when incorporating the physiological and the four IMU metrics on both legs.

out of the 31 metric combinations, while the 2s performed the best one time. Thus, the 3s window size consistently outperformed the other window sizes across metric combinations. The 1s, 5s, and 10s window sizes had poor performances in comparison (see Appendix A Figure A.3).

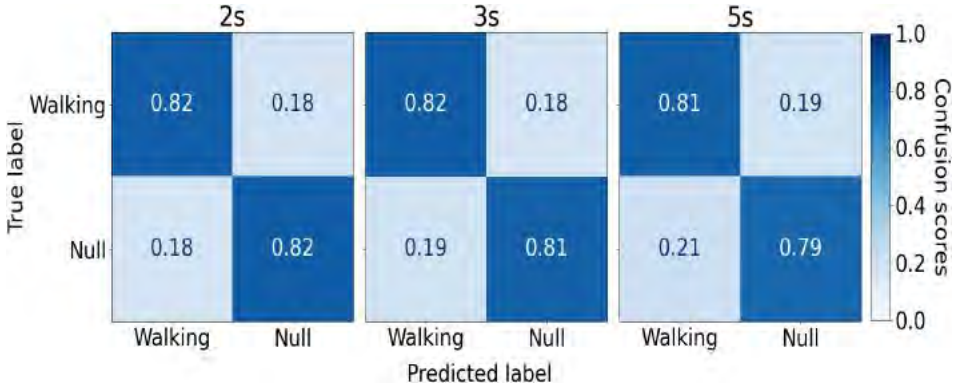


Figure 4.12: Gross motor task recognition confusion matrices when incorporating the physiological and four lower-body IMU metrics on both legs for the 2s, 3s, and 5s window sizes.

The incorporated metrics can impact the algorithm's performance; thus, using the 3s (i.e., best performing) window size, the algorithm was trained using 31 combinations of the metrics, as shown in Table 4.8. Interested readers can refer to the metric combinations results provided in Appendix A Table A.1 for the other window sizes.

The highest individual metric accuracy (80.80%) was achieved by the foot (F) IMU

Table 4.8: Gross motor task recognition accuracy (mean % (std. dev.)) by the 3s window size, and incorporated metrics aggregated across participants. NOTE: The highest accuracy and the corresponding sensor combination are highlighted in Bold, while the overall highest accuracy is in Blue.

No. of sensors	**Combination**	**Accuracy (%)**
1	Phy	61.80 (8.13)
	P	79.65 (8.20)
	T	80.33 (7.99)
	C	80.56 (7.86)
	F	**80.80 (7.81)**
2	Phy + P	79.98 (7.68)
	Phy + T	80.66 (8.34)
	Phy + C	81.13 (7.86)
	Phy + F	80.65 (8.33)
	P + T	80.53 (7.98)
	P + C	80.91 (7.83)
	P + F	**81.22 (7.80)**
	T + C	80.92 (8.10)
	T + F	81.19 (7.66)
	C + F	81.18 (7.62)
3	Phy + P + T	80.67 (8.10)
	Phy + P + C	80.99 (7.94)
	Phy + P + F	80.96 (7.91)
	Phy + T + C	81.02 (8.05)
	Phy + T + F	80.95 (7.82)
	Phy + C + F	80.95 (7.89)
	P + T + C	81.26 (7.76)
	P + T + F	81.00 (7.41)
	P + C + F	**81.33 (7.72)**
	T + C + F	81.16 (7.58)
4	Phy + P + T + C	81.21 (8.18)
	Phy + P + T + F	81.09 (7.91)
	Phy + P + C + F	80.99 (7.99)
	Phy + T + C + F	81.01 (7.91)
	P + T + C + F	**81.39 (7.54)**
5	**Phy + P + T + C + F**	**80.97 (7.89)**

metrics, while the physiological (Phy) metrics had the lowest accuracy (61.80%). The Wilcoxon signed-rank test revealed that physiological metrics' accuracy was significantly lower than the rest ($p < 0.01$, $2.17 <$ Cohen's $d < 2.36$). No significant difference in accuracy within the lower body (i.e., foot, calf (C), and thigh (T)) IMU metrics existed, but significant differences between the pelvis and all other lower body IMU metrics existed ($p < 0.05$, $0.08 <$ Cohen's $d < 0.14$) with a very small effect size. The algorithm's accuracy increased as the IMU sensors' displacement from the waist increased (i.e., pelvis < thigh < calf < foot), indicating that the lowest body point provided the most relevant features for identifying walking, followed by the subsequent lower body positions. The results are expected to vary for other gross motor tasks, especially upper-body tasks (e.g., lifting weights, and bending over).

Similar results were observed when combining two metrics, with the accuracy typically increasing when combining the lower body IMU metrics. The Phy + P combination's accuracy was significantly lower than the rest with a small effect size ($p < 0.05$, $0.08 <$ Cohen's $d < 0.16$), while the P + T combination was significantly lower ($p < 0.05$) than the Phy + C, P + F, T + F, and C + F combinations. No other differences were significant. The algorithm reached a saturation point when three or more metrics were combined, as none of the three or four metric combinations were significantly different, indicating that combining metrics beyond a certain limit can become excessively redundant and less meaningful for the evaluated scenario.

4.2.5.1 Discussion

Hypothesis $\mathbf{H_1^{GM}}$ predicted that the gross motor task recognition algorithm's accuracy will increase as the window size increases before reaching a point of diminishing returns. This hypothesis was fully supported. The 3s window size appears to be the optimal window size for this evaluation's gross motor task recognition, but may not be applicable for domains with varied gross motor tasks, including upper body tasks.

Hypothesis $\mathbf{H_2^{GM}}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the evaluated window sizes, which was supported. Although the algorithm achieved high sensitivity, its accuracy was expected to be $> 90\%$, as the number of detected tasks was limited and simpler (i.e., *Walking*). The unexpected decrease in accuracy can be attributed to confounded task labeling due to participants missing the scheduled *Walking* tasks frequently, because they were engaged with the other

four NASA-MATB composite tasks.

Overall, the algorithm's accuracy was higher with the IMU metrics, lower with the physiological metrics, and combining the IMU with the physiological metrics did not improve the accuracy. The evaluated supervisory domain's gross motor tasks were not complex (i.e., walking), but were detected with high sensitivity using only foot IMU metrics. The foot IMU can be complemented with additional IMU metrics (i.e., pelvis, calf, or thigh) in order to provide reasonable redundancy for the evaluated domain. Physiological metrics did not add any value to the supervisory domain; however, they can increase the recognition rate by improving context with other gross motor tasks. For example, combining IMU with heart rate can discriminate tasks that have similar motion patterns, but differ in intensity levels (e.g., *running* vs. *running with weights* [192]).

The algorithm's performance was comparable across the various lower-body IMU metrics and their combinations. This indiscernible change in performance can be attributed to detecting fewer gross motor tasks in the supervisory task environment. A discernible performance change across metrics is expected for domains that contain a larger number and more varied gross motor tasks (e.g., squatting, lifting an object, running, shoveling). A wide variety of gross motor tasks are required to better assess the algorithm's viability and the metrics' impact on the intended HRT domain. Additional IMU metrics (e.g., upper limb and shoulder positions) may be necessary to detect upper body gross motor tasks (e.g., lifting an object and shoveling).

4.2.6 Fine-Grained Motor Task Recognition

The fine-grained motor task recognition algorithm incorporated the Xsens IMU on the hands and wrists of both arms, as well as the two Myos' forearm IMUs and the 8-channel sEMGs. The algorithm employed up to eight CNNs, where each network extracted features pertaining to each metric's left and right arms. The CNN features were combined to predict one of the four fine-grained motor tasks: i) Joystick tracking, ii) Keyboard use, iii) Mouse use, and iv) Null for each window. Five window sizes ($t_w = \{1s, 2s, 3s, 5s, 10s\}$), and 15 metric combinations for the left, right, and both arms were investigated.

The fine-grained motor algorithm's accuracy when incorporating all four metrics for both arms, depicted in Figure 4.13, increased until the 3s window size (68.57%) and decreased to 62.09% at the 10s window size. The Friedman's test indicated a significant difference in accuracies between the window sizes ($\chi^2(4, 60) = 87.44, p < 0.01$). The

Wilcoxon signed-rank test found the 3s and 5s window size accuracies were significantly higher than all other window sizes with small to medium effect size ($p < 0.01$, $0.32 <$ Cohen's $d < 0.48$), while the 1s and 10s accuracies were significantly lower ($p < 0.01$) than the rest. The 2s window size's accuracy was significantly higher ($p < 0.01$) than 1s and 10s. The effect sizes between the 2s, 3s, and 5s accuracies were very small (Cohen's $d < 0.15$). No other differences were significant.

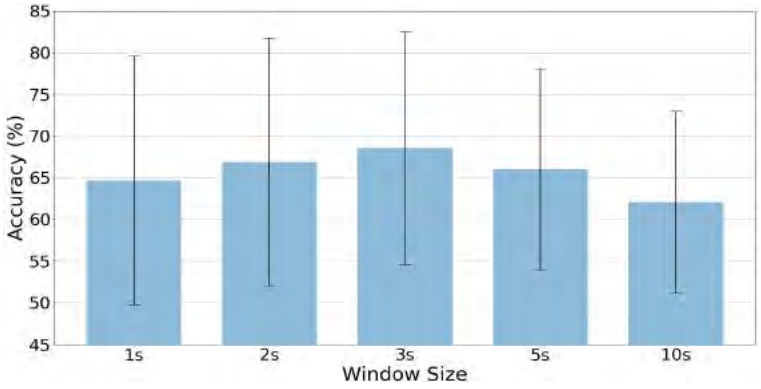


Figure 4.13: Fine-grained motor task recognition accuracy by window size with all four metrics from both arms.

The confusion matrices (see Figure 4.14) indicated that the 3s window size variant had less confusion for the *Keyboard* and *Mouse use* tasks, resulting in at least a 10% increase in the tasks' accuracies when compared to the 5s window size. All other tasks had similar confusions and accuracies. Therefore, the 3s window size's task-wise recognition rate was slightly better than the 5s window size, even though the overall accuracies were not significantly different.

The results can be generalized to the forty-five fine-grained motor metric combinations in that the 3s and 5s window sizes' accuracies were significantly higher than the rest, followed by the 2s window size. The 1s and 10s window sizes generally performed poorly. The 3s window size performed the best across 23 of the 45 combinations, as shown in Table 4.9, while the 5s window size performed the best seventeen times. The 1s, 2s, and 10s window sizes had poor results (see Appendix A A.4).

Understanding the metrics' impact on each arm is important, as it can minimize the number of wearable sensors used, and reduce the deep learning algorithm's trainable parameters. The algorithm was trained in three handedness variants using the 3s window

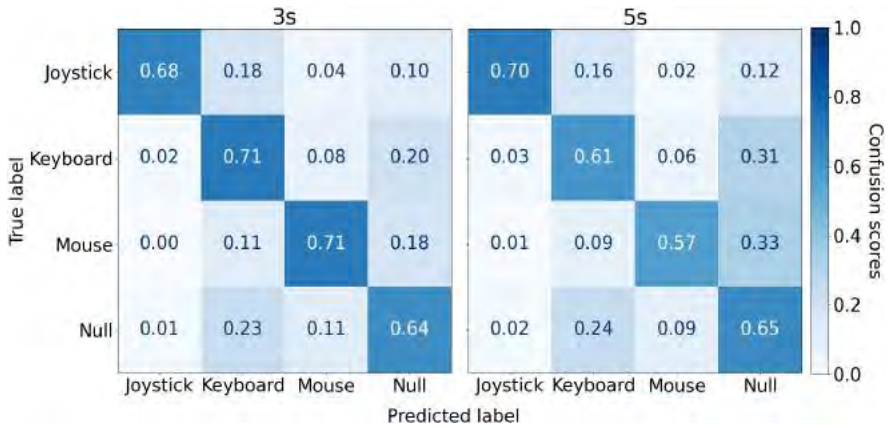


Figure 4.14: Fine-grained motor task recognition 3s and 5s window size confusion matrices when incorporating all four metrics from *both* arms.

Table 4.9: Frequency of the best-performing fine-grained motor task recognition algorithm variants by window size across the forty-five handedness and metric combinations.

Handedness	**Window size**				
	1s	**2s**	**3s**	**5s**	**10s**
Both	0	2	12	1	0
Left	0	1	1	11	2
Right	0	0	10	5	0
Overall	**0**	**3**	**23**	**17**	**2**

size. Incorporating metrics from *both* arms achieved the highest accuracy (68.57%), while the *left-only* metrics had the lowest accuracy (53.90%), and was significantly lower than the *right-only* and *both* metrics ($p < 0.01$, Cohen's $d > 1.17$). The *right-only* metrics' accuracy (64.62%) was significantly lower than *both* metrics ($p < 0.01$, Cohen's $d = 0.40$).

The handedness confusion matrices (see Figure 4.15) indicate that incorporating metrics from *both* arms had the least confusion and the best overall accuracy by tasks. The joystick tracking task had a better recognition rate with the *left-only* metrics, while the mouse use task had better accuracy with the *right-only* metrics, demonstrating that tasks can be arm-dependent. The Null task had the worst accuracy when incorporating *left-only* metrics, as it was often confused with keyboard and mouse use tasks. Keyboard use and Null tasks' recognition rates were higher with both hands, demonstrating that some tasks require features from both arms (i.e., the non-dominant arm may provide additional context even when not essential for the task).

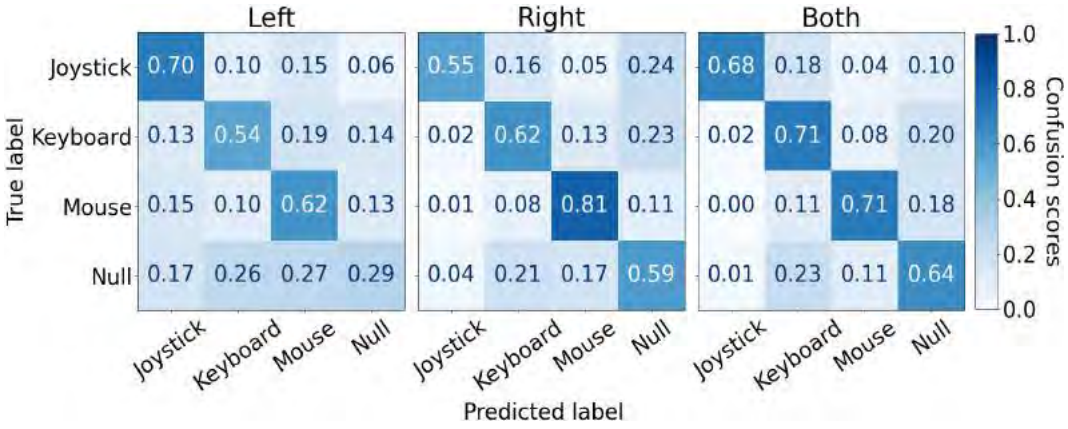


Figure 4.15: Fine-grained motor task recognition confusion matrices for all four metrics using the *Left-only*, *right-only*, and *both* arms for the 3s window size.

The specific metric combination can impact the algorithm's performance. A total of 15 combinations were evaluated individually and by combining two or more metrics (see Table 4.10) with the 3s window size and *both* handedness combination. Interested readers can refer to Appendix A Table A.2 for the other independent variable combinations.

The highest individual metric accuracy (68.56%) was achieved by the Xsens hand IMU metrics. The Friedman's test found a significant difference across the metrics ($\chi^2(14, 60) = 185.83, p < 0.01$). The Wilcoxon signed-rank test revealed that the Wrist (W) and Hand (H) IMU metrics' accuracies were significantly higher than all others ($p < 0.01$, Cohen's

Table 4.10: Fine-grained motor task recognition accuracy (mean % (std. dev.)) by the incorporated metrics using the 3s window size and *both* handedness combination aggregated across participants. The highest accuracy and corresponding sensor combination are highlighted in Bold, while the overall highest accuracy across all metrics combinations is highlighted in Blue.

No. of sensors	**Combination**	**Accuracy (%)**
1	F_{imu}	54.59 (18.51)
	H	**68.96 (9.26)**
	W	66.60 (10.66)
	F_{emg}	50.04 (16.49)
2	F_{imu} + H	65.37 (10.63)
	F_{imu} + W	65.05 (12.08)
	F_{imu} + F_{emg}	56.23 (20.33)
	H + F_{emg}	**69.29 (11.84)**
	W + H	68.51 (10.94)
	W + F_{emg}	68.47 (13.41)
3	F_{imu} + H + F_{emg}	66.21 (14.25)
	F_{imu} + W + H	67.31 (11.29)
	F_{imu} + W + F_{emg}	66.70 (15.01)
	W + H + F_{emg}	**70.04 (13.04)**
4	**F_{imu} + W + H + F_{emg}**	**68.57 (14.00)**

$d > 0.80$), while the Myos' sEMG metrics' was significantly lower ($p < 0.01$). The Myos' forearm IMU metrics' accuracy was significantly higher than the sEMG with a small effect size ($p < 0.01$, Cohen's $d = 0.26$).

Combining the Xsens hand IMU and the Myos' sEMG (H + F_{emg}) metrics achieved the highest accuracy (69.29%) when combining two metrics, while the lowest accuracy (56.23%) was recorded for the Myos' IMU and sEMG (F_{imu} + F_{emg}) combination. The Wilcoxon signed-rank test revealed that the F_{imu} + F_{emg} combination's accuracy was significantly lower than all other combinations with a medium effect size ($p < 0.01$, $0.52 <$ Cohen's $d < 0.78$). The H + F_{emg}, W + H, and W + F_{emg} combinations' accuracies were significantly higher, but with a smaller effect size than the F_{imu} + H, and F_{imu} + W combinations' accuracies ($p < 0.01$, $0.25 <$ Cohen's $d < 0.35$). No other differences were significant. The W + H + F_{emg} combination achieved the highest overall accuracy (70.04%), and was significantly higher than any of the other three metric combinations, but with a small effect size ($p < 0.01$, $0.22 <$ Cohen's $d < 0.28$).

Overall, the Xsens' IMU metrics' performance was significantly better ($p < 0.01$) than the Myo' forearm sEMG and IMU metrics. The algorithm's performance was the highest when incorporating the Xsens' hand or wrist IMU metrics across all combinations. The *Mouse* task had the highest recognition rate (82%). Combining sEMG metrics with hand and wrist metrics improved the recognition rate of most tasks, but reduced the *Mouse* task's recognition rate by 10%. Incorporating the Myos' forearm IMU metrics decreased the tasks' recognition rate across most metric combinations.

4.2.6.1 Discussion

Hypothesis $\mathbf{H_1^{FM}}$ predicted that the fine-grained motor task recognition algorithm's accuracy will increase, as the window size increases before reaching a point of diminishing returns, which was fully supported. Overall, the 3s window size performed the best across most metric and handedness combinations, with the 5s being a close second. Additionally, the 3s window size had a higher recognition rate and fewer misclassifications by tasks; therefore, the 3s is the suitable window size for detecting the evaluated supervisory tasks.

Hypothesis $\mathbf{H_2^{FM}}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one window size, which was not supported. The 70% maximum accuracy indicates that detecting fine-grained motor tasks with high sensitivity is difficult. The algorithm's poor performance can also be attributed to individual differences. Model-

ing individual human differences is difficult, but important. Customizing the algorithm for each individual can reduce the impact of individual differences and yield greater accuracy. The task recognition algorithm can be customized by retraining the deep learning model (i.e., updating the weights) with participant-specific data. Specifically, an individualized algorithm can be created by retraining a generalized algorithm (i.e., trained on every other participant's data) by combining the individual participant-specific data from their training session dataset and validating the individualized model on their trial session dataset (i.e., transfer learning). This approach will allow the algorithm to learn motion patterns that are exclusive to each participant.

Suboptimal task recognition due to different task completion times, especially across activity components is an open problem [18]. An ensemble learning algorithm that makes predictions over multiple fixed window sizes and fuses the predictions across the window sizes intelligently to detect the tasks may be required. An adaptive sliding window method that can expand and contract the window size based on the task may improve accuracy [170, 194, 195].

Overall, the Xsens' hand and wrist IMU metrics were the most important, as the metrics achieved comparable performance even individually. None of the individual Myo metrics (i.e., sEMG and forearm IMU) detected the tasks reliably. The algorithm's performance was significantly higher (up to 8%) when combining the hand and wrist IMU metrics with the Myos' forearm sEMG, rather than incorporating them individually. This metric combination appears to be ideal for detecting the supervisory domain's fine-grained tasks, but may not be preferred for all HRT domains.

The handedness analysis revealed that incorporating metrics from both arms was more important than the *right-only* metrics, followed by the *left-only* metrics. The *right-only* metrics were expected to outperform the *left-only* metrics, as over 80% of the global population prefer their right hand for complex tasks [69, 215]. However, the *both* handedness configuration results indicate that combining the left arm metrics affected the task recognition rate positively. This outcome may be attributed to i) about $10-20\%$ of the participants potentially being left-handed, ii) the evaluation's multi-tasking nature may have required participants to use both their arms for certain tasks (e.g., tracking and system monitoring), and iii) the algorithm may have benefited from the left arm's metrics providing context, even when the arm was not engaged in a task.

4.2.7 Tactile Task Recognition

The tactile task recognition algorithm incorporated inertial metrics provided by the Xsens sensors on the hands and the forearm 8-channel sEMG from the Myos to train a deep learning algorithm. The algorithm was trained to predict one of the four tactile tasks: i) Joystick tracking, ii) Keyboard stroke, iii) Mouse clicks, and iv) Null. Window sizes, $t_w = \{0.5s, 1s, 1.5s, 2s, 3s\}$, with a 50% overlap, were investigated for analyzing the window size's impact on the algorithm's performance. Smaller window sizes were used due to the tactile tasks' shorter durations.

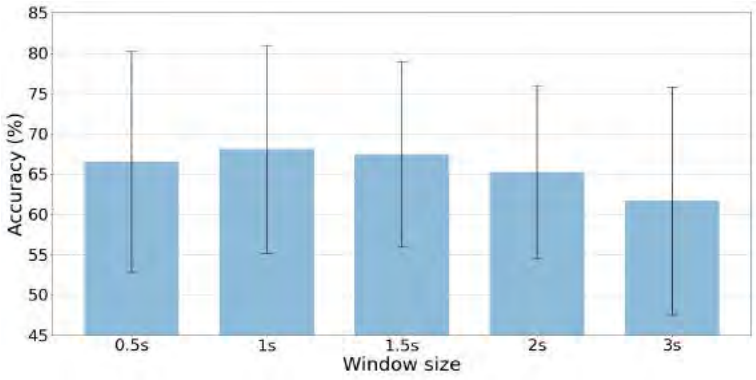


Figure 4.16: Tactile task recognition accuracy by window size for the IMU and sEMG metrics with both arms.

The tactile task recognition algorithm when incorporating the hand IMU and forearm sEMG metrics from both arms achieved the highest accuracy (68.06%) with the 1s window size, and decreased gradually to 61.68% for the 3s window size (see Figure 4.16). The Friedman's test indicated a significant accuracy difference between window sizes ($\chi^2(4, 60) = 24.09$, $p < 0.01$). The Wilcoxon signed-rank test indicated that the 1s window size's accuracy was significantly higher than all other window sizes ($p < 0.01$, $0.17 <$ Cohen's $d < 0.83$), while the 3s window size's accuracy was significantly lower than the rest ($p < 0.01$, $0.54 <$ Cohen's $d < 0.83$). The 1.5s window size's accuracy was significantly higher than the 2s and 3s ($p < 0.01$, $0.34 <$ Cohen's $d < 0.78$). The effect sizes between the 0.5s, 1s, and 1.5s accuracies were very small (Cohen's $d < 0.24$).

The tactile task recognition algorithm's confusion matrices incorporating the hand IMU and sEMG metrics from both arms for the 0.5s, 1s, and 1.5s window sizes are provided in Figure 4.17. The confusion matrices indicated that the 1.5s window size variant had more

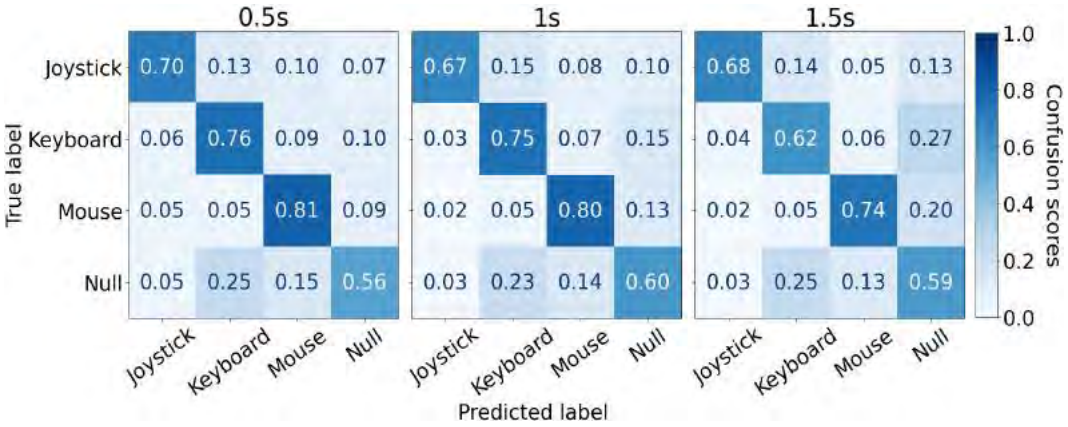


Figure 4.17: Tactile task recognition confusion matrices when IMU and sEMG metrics are incorporated on *Both* hands for 0.5s, 1s, and 1.5s window sizes.

confusion for the *Keyboard* (up to 13%) and *Mouse use* (up to 7%) tasks when compared to the 0.5s and 1s window sizes. The confusions and task accuracies between the 0.5s and 1s variants were similar with no significant differences. Additionally, Table 4.11 demonstrates that 1s window size performed the best for six out of the nine handedness and metric combinations, while the 1.5s window size performed the best 3 times. The 0.5s, 2s, and 3s window sizes had poor performances in comparison (see Appendix A Figure A.5).

Table 4.11: Frequency of the best-performing tactile task recognition algorithm variants by window size across the nine handedness and metric combinations.

Handedness	**Window size**				
	0.5s	**1s**	**1.5s**	**2s**	**3s**
Both	0	2	1	0	0
Left	0	3	0	0	0
Right	0	1	2	0	0
Overall	**0**	**6**	**3**	**0**	**0**

The algorithm was trained in three-handedness variants by incorporating the Xsens' hand IMU and the Myos' forearm sEMG metrics using the 1s window size. Incorporating metrics from *both* arms achieved the highest accuracy (68.06%), while the *left-only* metrics had the lowest accuracy (52.45%). The *both* arms metrics' were significantly higher than the *left-only* and *right-only* metrics ($p < 0.01$, $0.54 <$ Cohen's $d < 1.39$), while the *left-only*

metrics were significantly less accurate than the *right-only* metrics ($p < 0.01$, $0.85 <$ Cohen's $d < 1.39$). The handedness results with other window sizes and metric combinations are presented in Appendix A Table A.3.

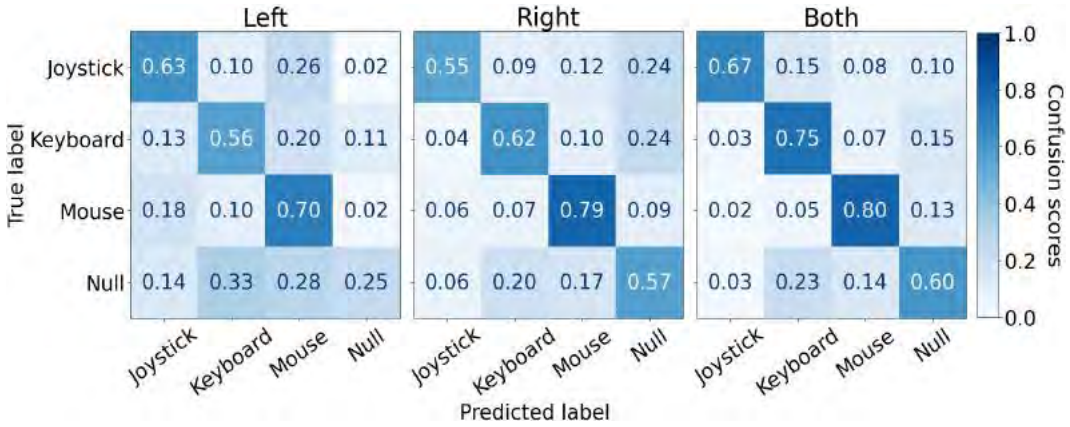


Figure 4.18: Tactile task recognition confusion matrices when hand IMU and sEMG metrics are incorporated on *Left-only*, *right-only*, and *both* arms using the 1s window size.

Comparing the confusion matrices by the handedness (see Figure 4.18) shows that incorporating metrics from *both* arms provided the best task recognition accuracy (i.e., the least confusions). Most tasks' recognition rates increased considerably when incorporating *both* metrics. The joystick tracking task had a better recognition rate with *left-only* metrics, while the mouse press task's recognition was better with the *right-only* metrics.

Table 4.12: Tactile task recognition accuracy (mean % (std. dev.)) by metrics for both arms with the 1s window size aggregated across participants. NOTE: The highest accuracy is highlighted in Bold.

Metrics	Accuracy
H	64.37 (7.83)
F_{emg}	51.69 (17.53)
H + F_{emg}	**68.06 (12.93)**

The multimodal combination H + F_{emg} achieved the highest accuracy (68.06%) and was significantly higher than when using only the hand IMU metrics, or the sEMG metrics ($p < 0.01$, $0.34 <$ Cohen's $d < 1.05$). Training the algorithm using only the F_{emg} metrics resulted in the lowest (51.69%) accuracy ($p < 0.01$, $0.92 <$ Cohen's $d < 1.05$).

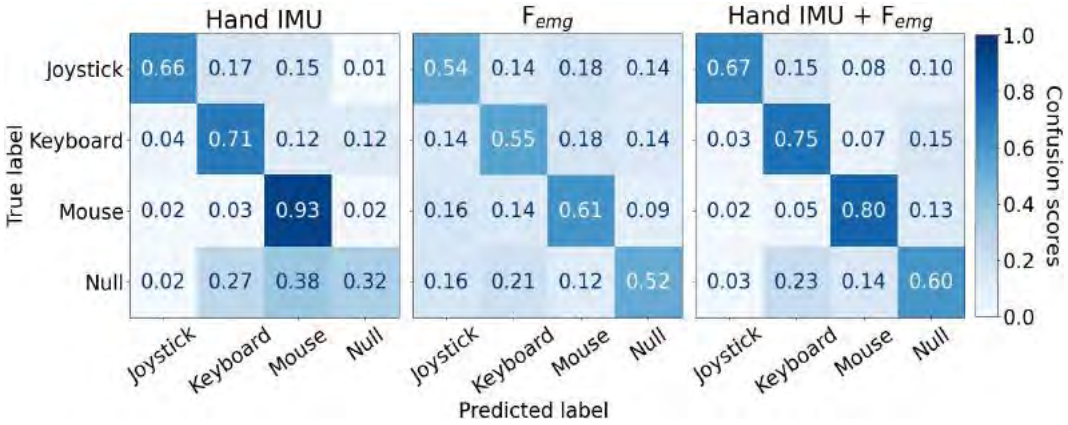


Figure 4.19: Tactile task recognition confusion matrices when only hand IMU, only F_{emg}, and Hand IMU + F_{emg} metrics are incorporated on both arms using the 1s window size.

An accuracy comparison by task and metric combination (see Figure 4.19) revealed that the multimodal combination had the overall best recognition rate across most tasks, and the sEMG-only combination performed the worst overall. The Hand IMU-only combination had the best recognition rate (93%) for *Mouse use* task; however, struggled to detect the *Null* task, which was often confused with the *Keyboard press* and *Mouse clicks* tasks. The algorithm trained with the Hand IMU-only metric may have inaccurately classified *Null* instances when the human's hands were resting on the keyboard or mouse, but were not pressing or clicking. Combining the hand IMU with the sEMG metric increased the *Null* task's recognition rate drastically (28%), indicating that the sEMG metric is extremely useful at distinguishing subtle tactile actions, especially when there is hardly any change in position and orientation (e.g., keyboard presses and mouse clicks).

4.2.7.1 Discussion

Hypothesis $\mathbf{H_1^T}$ predicted that the tactile task recognition algorithm's accuracy will increase with window size before reaching a point of diminishing returns, which was fully supported. Overall, the 1s window size performed the best across most metric and handedness combinations, with the 1.5s being a close second. The recommended window size is 1s given the supervisory domain's short-duration tactile tasks. An adaptive sliding window approach may be required for other domains.

Hypothesis $\mathbf{H_2^T}$ predicted that the tactile task recognition algorithm will detect tasks

with $\geq 80\%$ classification accuracy for at least one of the evaluated window sizes, which was not supported. The results suggested that detecting the extremely short-duration ($\leq$ 1s) tactile tasks with high accuracy was difficult.

Overall, the Xsens' hand IMU metric was the most valuable, as the metric achieved comparable performance even when incorporated individually. The Myos' sEMG metric was best utilized in conjunction with the Xsens' hand IMU metric, especially to distinguish between subtle tactile actions (e.g., *Null* and *Mouse press* tasks).

4.2.8 GNN Fusion Task Consolidation

The *Fusion* algorithm refined the components' atomic task detections by passing each individual algorithm's most recent task prediction scores as input to a GNN network to derive the atomic task predictions across components via joint optimization (see Chapter 3.6). The atomic tasks identified for each activity component are summarized in Table 4.13. The gross motor component had two atomic tasks, while the fine-grained motor, tactile, visual, and cognitive components had four atomic tasks each. The auditory and speech components had three and two atomic tasks, respectively. Thus, the GNN fusion algorithm consolidated a total of twenty-three atomic task detections and predicted the atomic tasks based on the seven activity components (i.e., one per component) at any given instance.

Table 4.13: Atomic tasks identified for each activity component when using the modified NASA MATB-II task environment.

Activity Component	**Atomic tasks**
Gross motor	Walking, Null
Fine-grained motor	Joystick tracking, Keyboard usage, Mouse usage, Null
Tactile	Joystick tracking, Keyboard stroke, Mouse clicks, Null
Visual	Tracking, Inspect, Locate, Null
Cognitive	Association, Evaluation, Conversation, Null
Auditory	COMM request, Walk ping, Null
Speech	COMM verbal response, Null

4.2.8.1 Experimental Design

The consolidated atomic detections can be fully correct, partially correct, or fully incorrect for a given instance. The standard accuracy metric used for the individual component

analyses does not account for partial correctness and may fail to capture the algorithm's performance correctly; therefore, it is modified to account for partial correctness. Accuracy per instance is calculated as the proportion of the predicted labels that are correct to the total number (i.e., predicted and true) of labels for a given instance. The *parital accuracy* is the average across all instances aggregated across participants (Equation 4.1) [72, 252].

$$Partial\ Accuracy = \frac{1}{N}\sum_{n=1}^{N}\frac{|\ \mathbf{x_n} \cap \hat{\mathbf{x_n}}\ |}{|\ \mathbf{x_n} \cup \hat{\mathbf{x_n}}\ |}, \tag{4.1}$$

where $\mathbf{x_n}$ is a list containing the true atomic labels across the seven components for an instance n, while $\hat{\mathbf{x_n}}$ is the list containing the seven predicted atomic task labels for the instance n.

Table 4.14: The individual algorithms and the corresponding window size and associated accuracy (mean % (std. dev.)) by component that were employed by the fusion algorithm for consolidating the atomic predictions.

Component	Algorithm	Window size	Accuracy
Cognitive	RF	15s	36.27 (5.03)
Speech	Deep learning	3s	93.57 (9.25)
Auditory	RF	10s	77.56 (14.82)
Visual	RF	60s	61.01 (11.86)
Gross motor	Deep learning	3s	80.97 (7.89)
Fine-grained motor	Deep learning	3s	68.57 (14.00)
Tactile	Deep learning	1s	68.06 (12.93)

Each individual algorithm's best-performing window size varied across components. The tactile task recognition algorithm performed the best for the 1s window size, while the visual task recognition algorithm performed the best for the 60s window size. Other components' best-performing window sizes and their corresponding accuracies are summarized in Table 4.14. The GNN fusion algorithm sourced each individual algorithm's task predictions from its corresponding best-performing window size as input to jointly optimize the atomic task detections across components.

The GNN fusion algorithm was evaluated using multiple window sizes $t_w = \{1s,\ 3s,\ 5s,\ 10s,\ 15s,\ 30s,\ 60s\}$ with a one-second stride (i.e., $t_s = 1$s) to inform the window size's impact on the GNN fusion algorithm's performance. The evaluated window sizes reflect the variability across the best-performing individual task component algorithms' window sizes,

with the maximum window size being 60s (i.e., the visual component's RF algorithm), while the stride duration was determined using the shortest duration possible to get an atomic task prediction (i.e., 1s for the tactile component).

4.2.8.2 Results

Overall, the fusion algorithm's partial accuracy for the 1s window size was 90.56% and slightly increased until the 60s (92.99%) (see Figure 4.20). The Friedman's test indicated a significant difference between window sizes ($\chi^2(6, 60) = 61.51, p < 0.01$). The Wilcoxon signed-rank test found that the 1s window size's partial accuracy was significantly lower than all other window sizes with a small effect size ($p < 0.01$, $0.08 <$ Cohen's $d < 0.38$). The 3s and 5s window sizes' accuracies were significantly lower than the 10s, 15s, 30s, and 60s window sizes with a small effect size ($p < 0.01$, $0.18 <$ Cohen's $d < 0.31$). No other differences were significant.

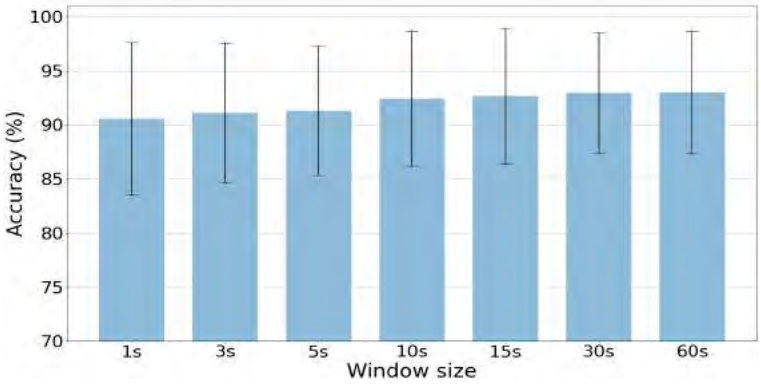


Figure 4.20: GNN fusion algorithm's partial accuracy % by window size aggregated across participants.

The fusion algorithm predicted seven atomic tasks at any given instance, each pertaining to one of the seven activity components; therefore, each component's accuracy improvements caused by the fusion algorithm can be compared against its corresponding best-performing individual algorithm's accuracy, as presented in Figure 4.21. Overall, the GNN fusion algorithm can detect the atomic tasks with $\geq$ 78% accuracy across all components. The GNN fusion's joint atomic task optimization improved the cognitive component's task detection the most (from 36% to 78%), followed by the tactile component (from 68% to 91%). The gross motor, fine-grained motor, visual, and auditory

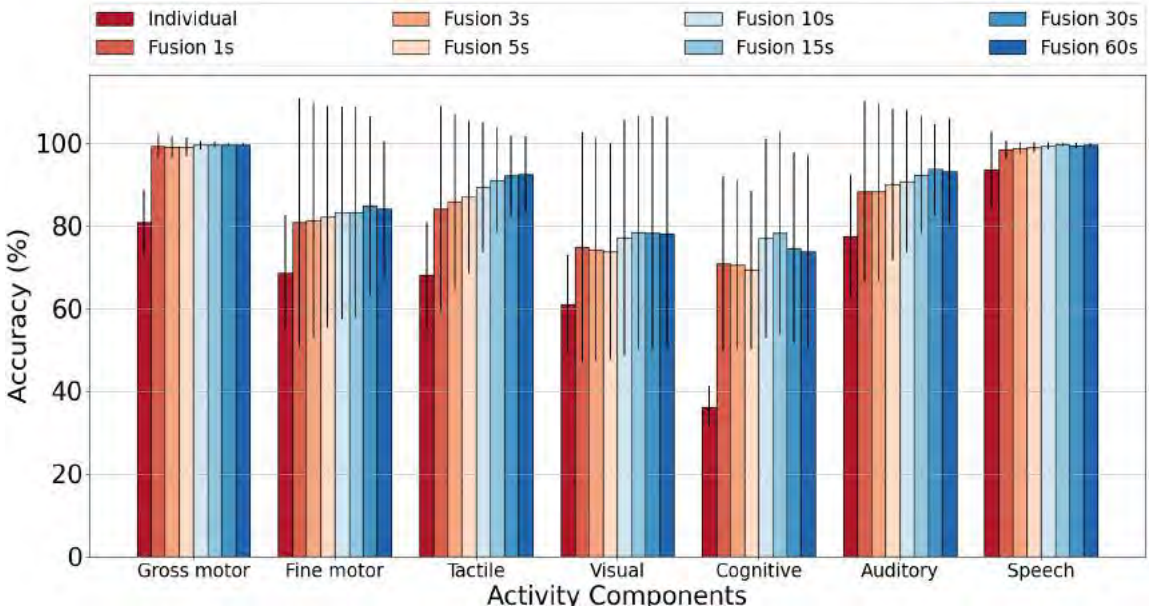


Figure 4.21: The accuracy (mean % (std. dev.)) comparisons between the individual algorithms and the GNN fusion algorithm by activity components for the evaluated window sizes. NOTE: Each component's individual algorithm's accuracy corresponds to its best-performing window size's accuracy.

increased by up to $16-19\%$, while the speech component's accuracy increased by 6%. The GNN fusion's joint optimization also increased the accuracy variability of activity components whose individual algorithm's accuracies were lower (i.e., fine-grained motor, tactile, visual, and cognitive) by at least 10%. The Wilcoxon signed-rank test indicated that the accuracies post GNN fusion's joint optimization were significantly higher than the corresponding individual algorithm's accuracies across all components ($p < 0.01$). Figure 4.21 indicates that the GNN fusion algorithm's performance was bottlenecked by the visual and cognitive components. The GNN fusion algorithm's 15s window size had the best overall performance, as it achieved the highest improvement for the bottleneck components (i.e., visual and cognitive), and $\geq 80\%$ accuracy across all the other components.

Each component's best-performing individual algorithm's confusion matrix is compared against the corresponding confusion matrix obtained using the 15s window size GNN fusion algorithm in order to analyze the accuracies by task across components pre (indicated as individual in the respective figures) and post (indicated as fusion in the respective figures) GNN fusion's joint optimization (see Figures 4.22, 4.23, 4.24, 4.25, 4.26, 4.27, 4.28). The

confusion matrices indicate that the *Null* atomic task was detected with near perfection (i.e., $\geq 97\%$) across all components. The GNN fusion's joint optimization increased the recognition rate of most tasks across components, with the cognitive component attaining the highest recognition rate improvement by tasks ($30-40\%$). The GNN fusion algorithm also reduced the recognition rate for a few tasks (e.g., fine-grained motor joystick and the visual inspect, locate, and tracking atomic tasks). These misclassifications can be attributed to the GNN fusion algorithm's bias toward the Null task, as it accounts for more than 80% of the data. Overall, the auditory, gross motor, and speech components' atomic tasks can be detected with $\geq 80\%$ accuracy post GNN fusion's joint optimization, while the cognitive, tactile, and fine-grained motor achieve either $\geq 80\%$ or $\sim 80\%$ for most tasks. The visual component's non-Null tasks' recognition rates did not improve over the individual algorithm's accuracy. Interested readers can refer to Appendix A Chapter A.6 for the rest of the GNN fusion algorithm's window sizes' confusion matrices.

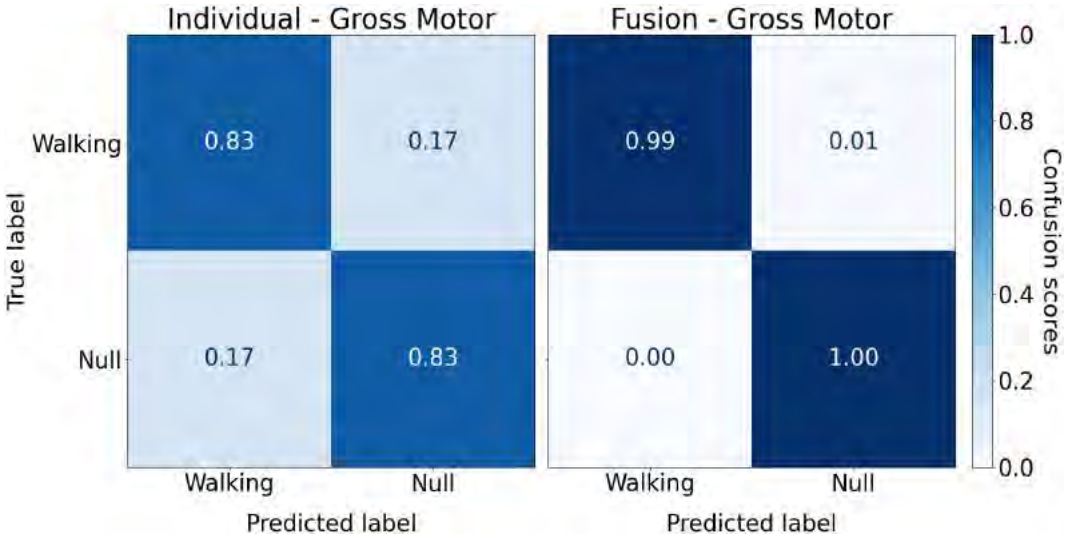


Figure 4.22: Gross motor component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (15s window size).

4.2.8.3 Discussion

The GNN fusion algorithm had a high sensitivity across all window sizes; however, the partial accuracy may have been artificially inflated due to non-uniform data distribution.

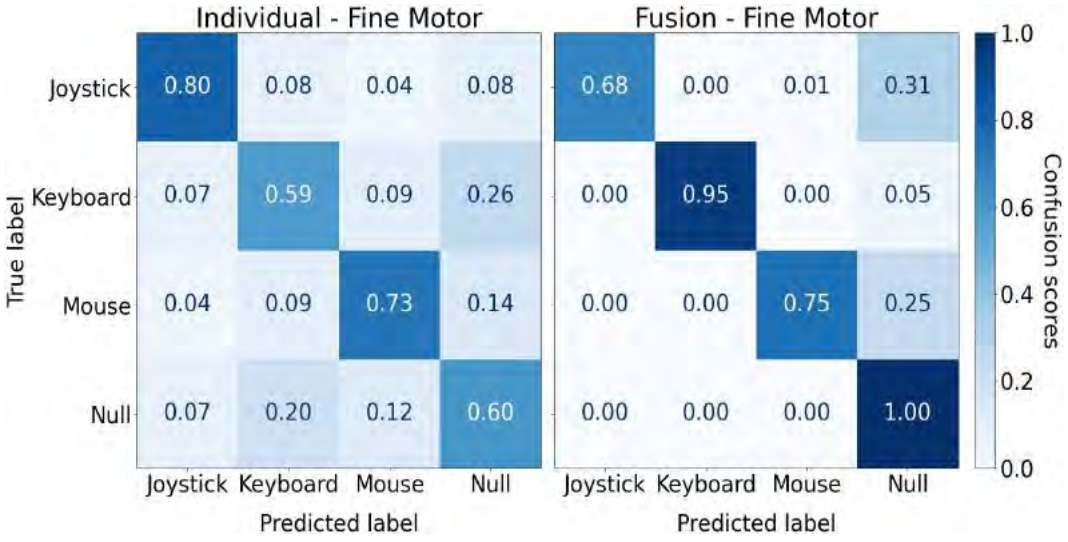


Figure 4.23: Fine motor component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (15s window size).

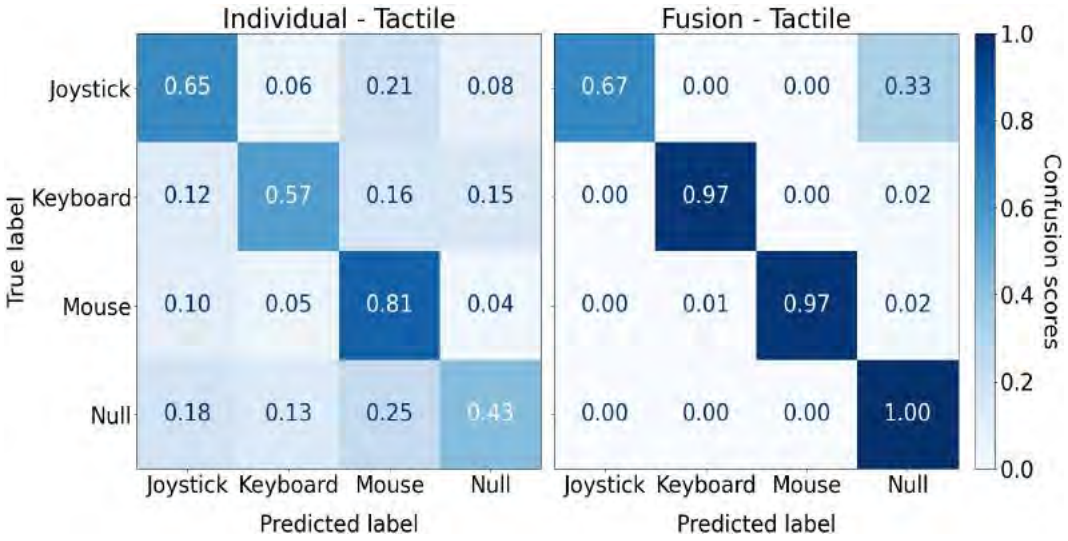


Figure 4.24: Tactile component's confusion matrix for its best-performing individual algorithm (1s window size) vs. GNN fusion algorithm (15s window size).

Figure 4.25: Visual component's confusion matrix for its best-performing individual algorithm (60s window size) vs. GNN fusion algorithm (15s window size).

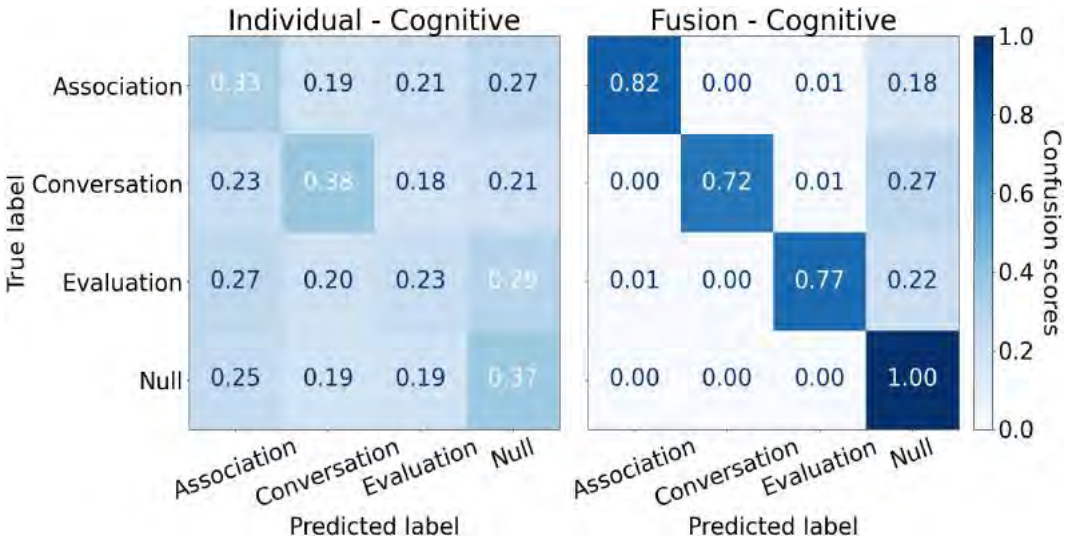


Figure 4.26: Cognitive component's confusion matrix for its best-performing individual algorithm (15s window size) vs. GNN fusion algorithm (15s window size).

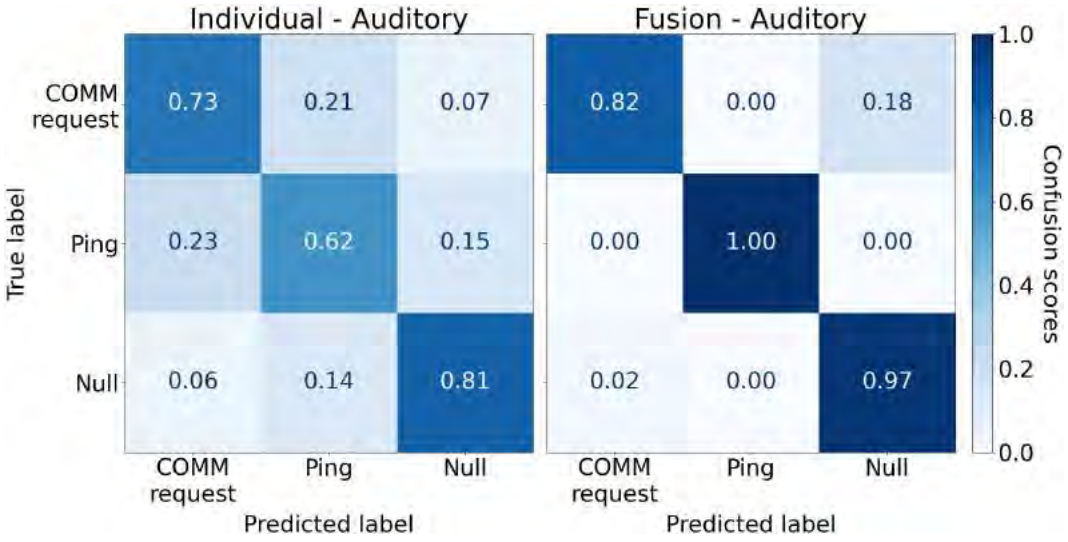


Figure 4.27: Auditory component's confusion matrix for its best-performing individual algorithm (10s window size) vs. GNN fusion algorithm (15s window size).

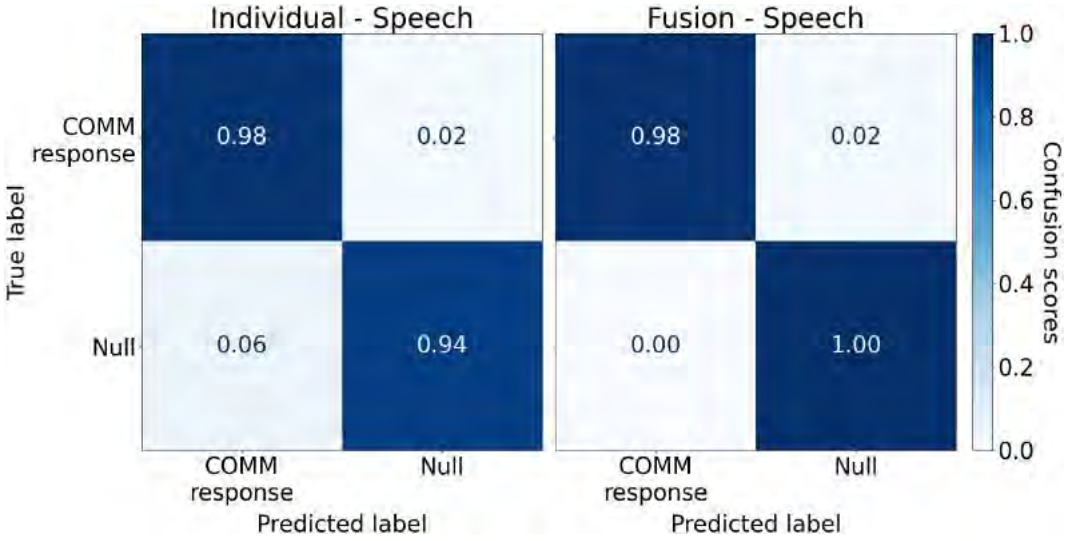


Figure 4.28: Speech component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (15s window size).

All the activity components followed a long-tailed distribution, where certain tasks, typically the *Null* task, account for more than 80% of the data, while all the other tasks were under-represented. The *Null* tasks were far easier to detect across components, which when consolidated across all seven components may have biased the fusion algorithm, resulting in inflated partial accuracy. Nevertheless, the GNN fusion algorithm is a viable candidate to consolidate the atomic task predictions. The 15s window size is the recommended window size, as it had the best performance across all components.

Hypothesis $\mathbf{H_3}$ predicted that the GNN fusion algorithm's joint task optimization will improve the atomic task detection accuracy to $\geq 80\%$ across all seven components, which was partially supported. The GNN fusion improved the atomic recognition rate for most tasks across components by leveraging the underlying graphical structure and adjacency correlation matrix to improve the components' task detections. However, the high accuracies can be attributed to the limited number of atomic tasks per component. A peer-based evaluation with more tasks per component is required to further evaluate the GNN fusion algorithm's ability to consolidate the atomic task detections across multiple domains.

4.2.9 Composite and Concurrent Task Recognition

The TCN-based *Composite and Concurrent* task recognition algorithm (described Chapter 3.7) incorporated the atomic task time series $\mathbf{X}$ as input to predict five composite tasks: i) Tracking, ii) System monitoring, iii) Resource management, iv) Communication request, and v) Communication response. The five composite tasks correspond to the evaluated NASA-MATB tasks, where the communication composite task was split into two subtasks, communication request and communication response, in order to model the radio request and any verbal response.

4.2.9.1 Experimental Design

The TCN-based algorithm detected the concurrent composite tasks (i.e., ≥ 1 composite tasks) by predicting the probability of each composite task for a given atomic task time series; therefore, the algorithm's predictions can be fully correct, partially correct, or fully incorrect. Two dependent variables, *exact match ratio* and *partial accuracy*, along with multi-label confusion matrices were used to evaluate the TCN-based algorithm's performance. The exact match ratio is the multi-label extension of the standard accuracy

metric, where an instance is deemed correct if and only if the algorithm predicts all the composite tasks that are present, and rejects all the composite tasks that are absent for a given atomic task time series instance (Equation 4.2) [72, 252].

$$\textit{Exact Match Ratio} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y_n} = \hat{\mathbf{y_n}}\,, \tag{4.2}$$

where $\mathbf{y_n}$ is a list containing the true composite tasks for the instance n, while $\hat{\mathbf{y_n}}$ is the list containing the predicted composite tasks for the time series instance $\mathbf{X_n}$.

The primary limitation of the exact match ratio metric is that it does not distinguish between completely incorrect and partially incorrect. Therefore, the partial accuracy (Equation 4.3) metric, modified to account for the composite tasks, is used as a second dependent variable to evaluate the TCN-based algorithm's performance.

$$\textit{Partial Accuracy (composite)} = \frac{1}{N}\sum_{n=1}^{N} \frac{\mid \mathbf{y_n} \cap \hat{\mathbf{y_n}} \mid}{\mid \mathbf{y_n} \cup \hat{\mathbf{y_n}} \mid}\,, \tag{4.3}$$

The multi-label confusion matrices account for task concurrency (i.e., ≥ 1 composite tasks for a given instance) by plotting the confusion matrix for each composite task. Each confusion matrix's top-left tile indicates the corresponding composite task's true negative instances predicted by the algorithm, the top-right tile indicates false positives, the bottom-left indicates false negatives, and the bottom-right corresponds to true positives. The multi-label confusion matrices' values are normalized to remain between 0 to 1. Ideally, the TCN algorithm must predict a composite task if it is present (i.e., true positive) and reject if it is absent (i.e., true negative) for a given instance; therefore, high (≥ 0.8) true positives and true negatives are expected. Similarly, the algorithm must not predict a composite task that was not present (i.e., false positive), and not reject a composite task that was present (i.e., false negative); therefore, the false positives and false negatives are expected to be very low.

The input time series $\mathbf{X}$'s temporal duration was varied in several window sizes $t_w = \{1s, 3s, 5s, 10s, 15s, 30s, 60s\}$ with a one-second stride (i.e., $t_s = 1s$) to inform its impact on the TCN algorithm's performance. The evaluated window sizes reflect the range of window sizes examined for the fusion algorithm, while the stride duration was chosen to match the shortest window size incorporated across all components (i.e., 1s for the tactile).

4.2.9.2 Results

Overall, the algorithm's exact match ratio for the 1s window size was 78.64% and gradually increased until the 60s window size (88.21%) (see Figure 4.29). The exact match ratio's std. dev. decreased from 7.77% for the 1s window size to 6.09% for the 15s window size, and started increasing to 8.44% and 9.87% for the 30s and 60s window sizes, respectively. This trend in std. dev. indicates that the algorithm's uncertainty decreased with the increase in the temporal window size before reaching a threshold beyond which the algorithm is more accurate, but less precise.

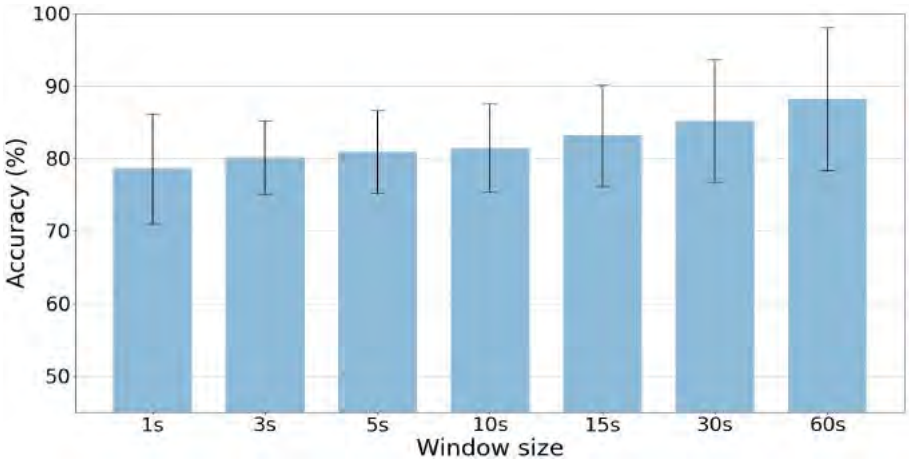


Figure 4.29: TCN composite and concurrent task recognition algorithm's exact match ratio % by window size aggregated across participants.

The Friedman's test indicated a significant difference between window sizes ($\chi^2(6, 60) = 255.70, p < 0.01$). The Wilcoxon signed-rank test found that the 60s window size's exact match ratio was significantly higher than all other window sizes ($p < 0.01$, $0.32 <$ Cohen's $d < 1.08$), while the 1s window size's was significantly lower than the rest with a large effect size ($p < 0.01$, $0.23 <$ Cohen's $d < 1.08$). The 10s window size's exact match ratio was significantly higher than the 3s and 5s ($p < 0.01$, $0.09 <$ Cohen's $d < 0.23$), while the 5s window size's exact match ratio was significantly higher than the 3s ($p < 0.01$, Cohen's $d = 0.14$). The 15s window size's exact match ratio was significantly higher than the 3s, 5s, and 10s ($p < 0.01$, $0.26 <$ Cohen's $d < 0.49$), while the 30s window size's exact match ratio was significantly higher than the 3s, 5s, 10s, and 15s ($p < 0.01$, $0.26 <$ Cohen's $d < 0.72$).

The algorithm's partial accuracy for the 1s window size was 77.16% and continued

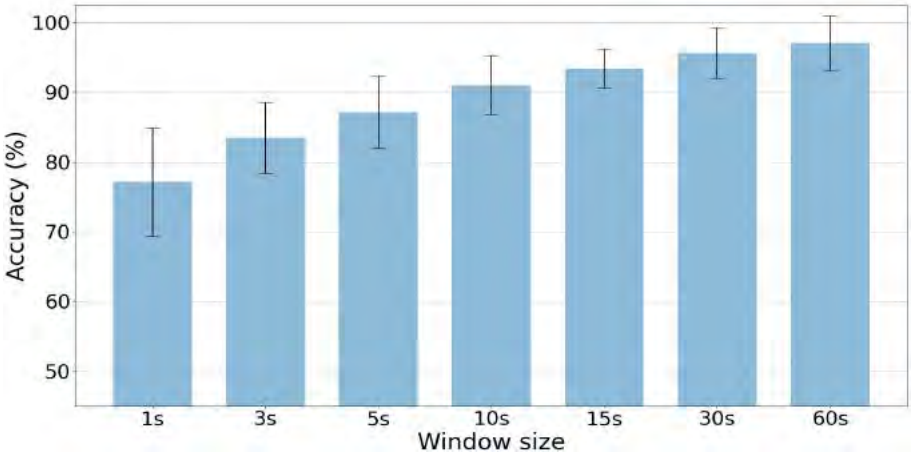
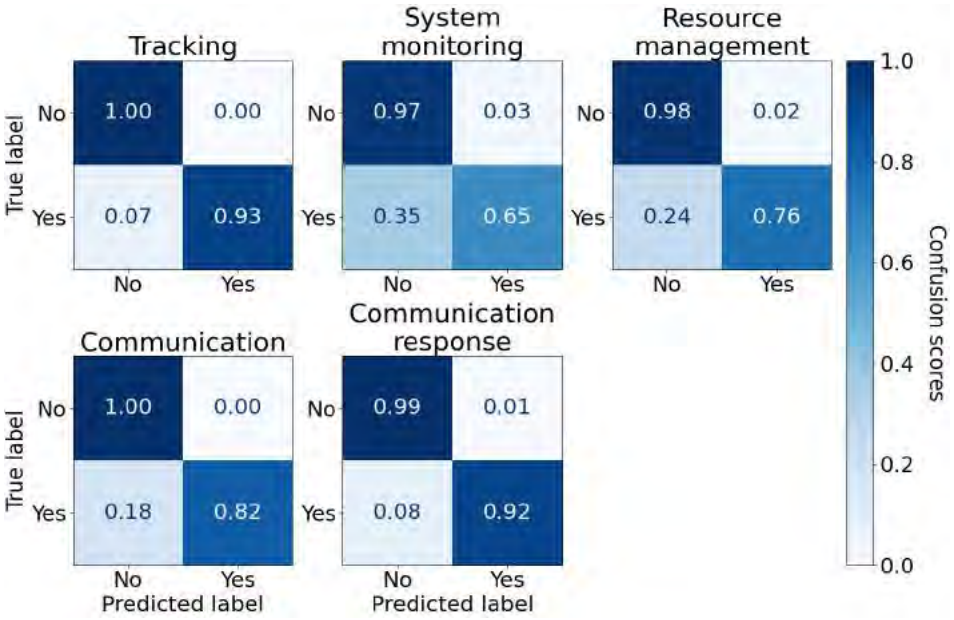


Figure 4.30: TCN composite and concurrent task recognition algorithm's partial accuracy % by window size aggregated across participants.

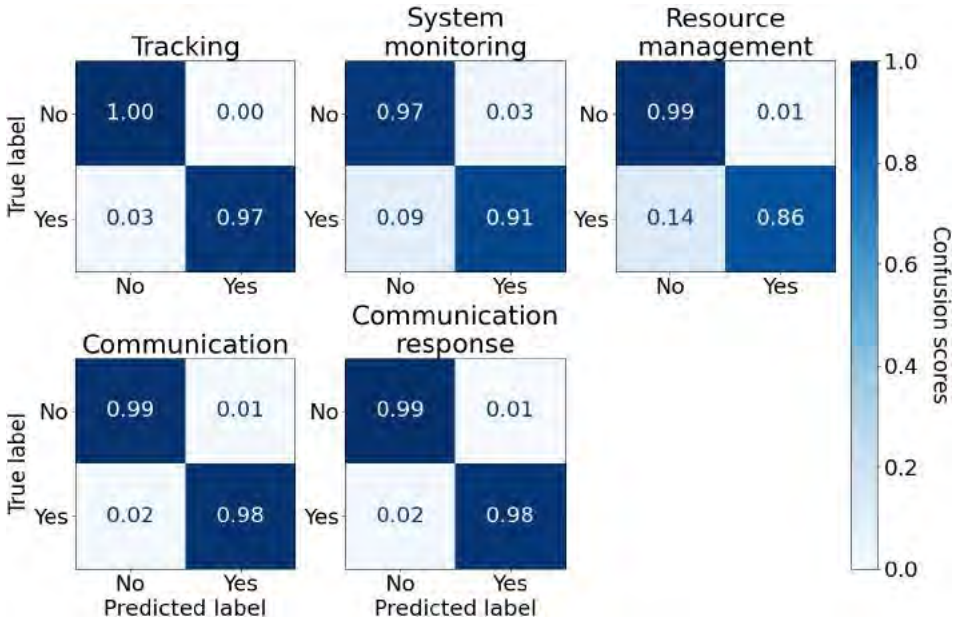
to increase until the 60s window size (97.07%) (see Figure 4.30). Further, the partial accuracy's std. dev. decreased gradually from 7.75% for the 1s window size to 2.77% for the 15s window size, and slightly increased to 3.58% and 3.91% for the 30s and 60s window sizes, following a trend similar to the algorithm's exact match ratio. The Friedman's test indicated a significant difference in partial accuracy between window sizes ($\chi^2(6, 60) = 369.05, p < 0.01$). The Wilcoxon signed-rank test found that the 60s window size's partial accuracy was significantly higher than all other window sizes ($p < 0.01$, $0.42 <$ Cohen's $d < 3.22$), while the 1s window size's was significantly lower than the rest with a large effect size ($p < 0.01$, $0.96 <$ Cohen's $d < 3.22$). The 10s window size's partial accuracy was significantly higher than the 3s and 5s ($p < 0.01$, $0.81 <$ Cohen's $d < 1.59$), while the 5s window size's partial accuracy was significantly higher than the 3s ($p < 0.01$, Cohen's $d = 0.70$). The 15s window size's partial accuracy was significantly higher than the 3s, 5s, and 10s ($p < 0.01$, $0.59 <$ Cohen's $d < 2.21$), while the 30s window size's partial accuracy was significantly higher than the 3s, 5s, 10s, and 15s ($p < 0.01$, $0.70 <$ Cohen's $d < 2.92$).

The results indicated that the algorithm's performance in terms of both exact match ratio and partial accuracy increased with the temporal window size. The multi-label confusion matrices were analyzed for 1s, 15s, and 60s window sizes (see Figure 4.31) in order to understand the differences in performance by tasks between the window sizes. The 1s and 60s window sizes represented the algorithm's extremes in performance, while the 15s window size presented the trade-off in performance between the two extremes. Other

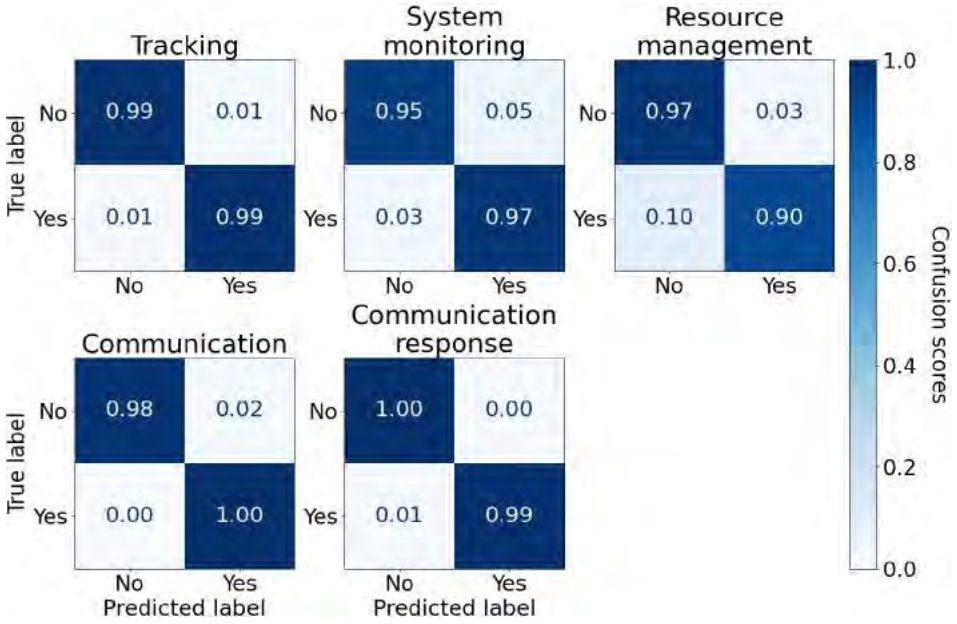
window sizes had intermediate performances and the confusion matrices are provided in Appendix A Figure A.13. The algorithm detected the composite tasks' absence (i.e., true negatives) with high accuracy ($> 95\%$) across all three window sizes. The 1s window size's true positive rates were $10 - 30\%$ worse than the other window sizes for most tasks. The 15s and 60s window sizes had comparable true positives across four of the five composite tasks. The 15s window size's true positive rates were slightly worse for system monitoring (6%) and resource management (4%) composite tasks.



(a) 1s window size



(b) 15s window size



(c) 60s window size

Figure 4.31: TCN composite and concurrent task recognition algorithm's multi-label confusion matrices by tasks for the 1s, 15s, and 60s window sizes.

4.2.9.3 Discussion

Hypothesis $\mathbf{H_4}$ predicted that the TCN composite and concurrent task recognition algorithm's accuracy will increase with the window size before reaching a point of diminishing returns. This hypothesis was only partially supported, as the exact match ratio (%) and partial accuracy did not reach a saturation point and continued increasing. The results indicate that the TCN's dilated causal convolutions benefit from increasing the atomic task time series' window size, providing the algorithm with more temporal context; however, the improvement in algorithm performance comes with a trade-off. The TCN-based deep learning algorithm's trainable parameters are directly proportional to the temporal window size, requiring increased computational resources to train and run the algorithm for larger window sizes. The 15s window size is recommended over the 60s for detecting composite tasks in a supervisory-based domain, as it is shorter in duration, contains a lower number of trainable parameters, and almost mirrors the 60s window size's performance. Larger window sizes may be utilized when accuracy is of paramount importance.

Hypothesis $\mathbf{H_5}$ predicted that the TCN algorithm will detect composite tasks occurring concurrently with $\geq 80\%$ accuracy, which was supported for the $\geq 15s$ window sizes. The algorithm's high recognition rate can be attributed to the limited number of composite tasks investigated. A peer-based evaluation with more composite tasks is required to further evaluate the TCN algorithm's ability to detect composite tasks that occur concurrently.

4.3 Summary

Identifying the appropriate window size for each algorithm informs how the metrics must be segmented, such that the features extracted (i.e., handcrafted or via deep learning) are representative of the tasks being detected. Hypothesis $\mathbf{H_1}$ focused on determining the optimum window size for the incorporated metrics in order to realize the individual algorithms' full potential. The gross motor, fine-grained motor, tactile, and speech task detection algorithms required smaller windows ($< 5s$), while the cognitive, visual, and auditory algorithms required larger window sizes ($\geq 10s$). Environments with rapid task switching (e.g., overload condition) will negatively affect the metrics that require larger window sizes and, thereby the algorithms that incorporate such metrics. Overall, metrics that require smaller windows are preferred for atomic task recognition, due to their ability to identify changes within a short duration, such that the atomic tasks can be detected

before task switching. Thus, further evaluation of the larger window size metrics is required to assess their viability.

It is important to determine the incorporated metrics' ability to detect the tasks reliably across multiple activity components; thus, hypothesis $\mathbf{H_2}$ focused on evaluating the individual algorithms' capability to detect tasks with high sensitivity ($\geq 80\%$). $\mathbf{H_2}$ is fully supported for only the gross motor and speech task components, partially supported by the auditory component, and is not supported for the fine-grained motor, tactile, visual, and cognitive task detection components. Additionally, fine-grained motor and tactile tasks are susceptible to individual differences; therefore, algorithm customization may be required to improve their sensitivity. The visual and cognitive task detection analysis determined that the incorporated metrics are less responsive to reliably detect tasks in a dynamic, multi-tasking environment (i.e., switching tasks frequently).

The individual algorithms' atomic task detections are not reliable for at least half of the components. The GNN fusion algorithm bridged this gap by facilitating indirect atomic task inference for components with subpar accuracy via joint optimization. The GNN fusion algorithm obtained the most recent atomic task prediction from each component's individual algorithm by looking back over a $15s$ window and optimized those predictions across components, thereby elevating the components' atomic task recognition sensitivity level to $\geq 80\%$. The TCN algorithm detected the concurrent, composite tasks with high sensitivity and low uncertainty, given sufficient temporal context about the atomic tasks; however, increasing the temporal context beyond a threshold can lead to diminishing returns.

Robots need a holistic understanding of a task's various activity components in order to identify what task(s) humans are executing. An important aspect of such understanding is detecting the human teammates' tasks across components. The developed individual task recognition algorithms, along with the GNN fusion and TCN algorithms are viable candidates for detecting atomic and concurrent, composite tasks across components. These algorithms are suitable to be incorporated into an adaptive HRT architecture that allows robots to adapt to the teammate's state, but with a few limitations. The algorithms' capabilities can be improved by customizing the algorithm to account for individual differences, and incorporating adaptive or ensemble sliding window methods to detect tasks of varying lengths. The task detection algorithms will need to be evaluated in an uncertain, dynamic peer-based task environment with a wide range of tasks in order to assess its viability across domains.

Chapter 5: Peer-Based Experimental Analysis

HRTs involve humans and robots collaborating to achieve tasks under various environmental conditions, requiring the robot teammates to adapt autonomously to a human teammate's state. An essential element of such adaptation is the robots' ability to infer the tasks performed by their human teammates. The prior supervisory evaluation (see Chapter 4) demonstrated the multi-dimensional task recognition algorithm's ability to detect atomic tasks across components, and detect concurrent, composite tasks, but was limited by the number of included tasks and the evaluated environment. For example, the supervisory evaluation's gross motor component only incorporated a walking task. Reliable HRT task recognition requires the algorithm to detect a broad range of tasks under various conditions, especially in unstructured, dynamic environments (e.g., post-tornado disaster response).

A peer-based human subjects evaluation was designed to manipulate a wide variety of composite tasks across components. A task hierarchy (see Chapter **??**) was developed to define how different tasks relate to one another and to specify how these tasks relate to the different activity components. Mission tasks are comprised of multiple composite and atomic tasks, while composite tasks can be further subdivided into multiple sub-composite and atomic tasks across components. An example mission task is *Clearing a pharmacy of controlled substances*, which consists of multiple composite tasks, such as *Searching for an item*. This composite task can be further subdivided into sub-composite and atomic tasks (e.g., walking, visual scanning). Mission tasks that have similar characteristics can be grouped together, *mission groups*. For instance, *Clearing a pharmacy of controlled substances* and *Clearing a pawnshop of dangerous weapons* are different mission tasks, but share similar characteristics and goals.

Human state estimation algorithms that are developed in controlled experimental environments struggle to translate to real-world problem domains; thus, the presented experimental design emphasized ecological validity. Ecological validity is defined as the extent to which results of an evaluation can be generalized to real-life settings [244]. This evaluation incorporated ecological validity by designing realistic disaster response and civil-support mission tasks with realistic human-robot teaming dynamics.

5.1 Experimental Design

A mixed-subjects peer-based user evaluation was designed to assess the multi-dimensional task recognition algorithm's ability to detect the concurrent, composite tasks performed by human teammates in peer-based HRTs operating in unstructured, dynamic environments. This evaluation served three core purposes that required collecting data incorporating a broad range of human tasks to facilitate the development of 1) the multi-dimensional task recognition algorithm, 2) a multi-dimensional workload estimation algorithm to make accurate estimates for known and unknown mission tasks, and 3) a short-term and long-term workload predictions algorithms. This evaluation manipulated mission tasks, task density, workload ordering, and training session type as independent variables. Prior user evaluations focused on eliminating learning effects, and trained participants on all mission tasks to be evaluated. This comprehensive training eliminated the existence of unknown mission tasks and made it impossible to evaluate a workload model's ability to generalize to unknown mission tasks. This evaluation introduced the independent variable *training session type*, which restricted participants to only training on a subset of mission tasks.

The task environment consisted of a simulated first response scenario, where the HRT responded to the aftermath of a tornado. Participants were paired with a Pioneer 3DX robot, and the team was tasked with performing six *mission* tasks: clearing a pharmacy of controlled substances, searching an area for suspicious items, sampling hazardous powder substances, clearing debris from a road, clearing a pawnshop of dangerous weapons, and sampling hazardous liquid substances.

This evaluation was conducted over two days. Participants completed a one-hour training session on the first day, where they were trained to perform a subset of the mission tasks. During the second trial day, the participants completed the full 70-minute trial composed of seven consecutive 10-minute mission tasks.

Designing a user evaluation that achieves all of these goals is a non-trivial process. First, this evaluation's hypotheses, independent and dependent variables relevant to this dissertation are presented. A high-level overview of the task environment and the HRT is provided, followed by an explanation of each of the three experimenter's roles. Next, a detailed discussion of each mission task is presented, alongside the associated task decomposition. This task decomposition informed how the mission tasks were decomposed into composite and atomic tasks, as well as how IMPRINT Pro was utilized, prior to conducting the evaluation, to verify that the overall workload and each workload component were

properly manipulated. Lastly, the participant demographics are presented.

5.1.1 Hypotheses

This evaluation supported three dissertations. This dissertation focuses on developing a multi-dimensional task recognition algorithm capable of accurately detecting tasks across components and using those detections to detect concurrent, composite tasks performed by human teammates in uncertain, dynamic environments. The second dissertation focuses on developing a meta-learning multi-dimensional workload estimation algorithm capable of accurate estimates for unknown mission tasks across components [250]. Lastly, the third dissertation focuses on developing workload prediction algorithms to forecast workload over both short-term (i.e., 30 seconds) and long-term (i.e., 10 minutes) time horizons. It is important to note that many of the decisions for this evaluation's experimental design are influenced by all three dissertations, as a single human-subjects evaluation was created to collect results to support all three dissertations.

Three hypotheses are formed to evaluate the multi-dimensional task recognition algorithm. Hypothesis $\mathbf{H_1}$ states that each individual task detection algorithm will detect the peer-based atomic tasks with $\geq 80\%$ classification accuracy for at least one of the analyzed window sizes. Hypothesis $\mathbf{H_2}$ predicts that the GNN fusion algorithm will result in highly sensitive ($\geq 80\%$ accuracy) atomic task detection by jointly optimizing the individual algorithms' atomic task predictions across components. Hypothesis $\mathbf{H_3}$ states that the TCN task recognition algorithm will detect concurrent composite tasks with high sensitivity ($\geq 80\%$ accuracy).

5.1.2 Independent Variables

The mixed-subjects evaluation consisted of two within-subjects independent variables (i.e., task, task density) and two between-subjects independent variables (i.e., training session type, workload ordering), shown in Table 5.1. The within-subjects *task variable* corresponds to the seven mission tasks a participant performed during the trial session. These mission tasks mirror realistic tasks performed by different first response groups (e.g., police, fire, civil support) and belong to four mission task groups: clearing, sampling, debris, and searching, shown in Table 5.2. The clearing group's mission tasks required the participant and the robot to independently search an area for known objects (e.g., pill bottles,

fake hand guns) that needed to be collected. There were two mission tasks in the clearing group, the *Pharmacy task* and the *Pawnshop task*. The debris group had one mission task (i.e., the *Debris task*), during which the participant and the robot collaborated to clear a path through a pile of debris. Mission tasks in the sampling group required the participant to collect samples of various substances by following detailed step-by-step procedures provided by the robot. This procedure followed strict guidelines for maintaining safe and sterile sampling procedures by published government standards [167]. The sampling group consisted of two mission tasks: the *Liquid Sampling task*, and the *Solid Sampling* task. Both mission tasks consisted of a similar procedure, but required the participant to use slightly different tools to gather liquids vs. solid substances. Lastly, the searching group consisted of a single task (i.e., the *Search task*) that required searching an area for dangerous, or suspicious objects.

Table 5.1: The independent variables for the peer-based evaluation.

Type	**Variable**	**Values**
within-subjects	Mission tasks	Pharmacy, Pawnshop, Debris, Search, Solid sampling, Liquid sampling.
	Workload (i..e, task density)	UL, NL, OL
between-subjects	Training session type	Type 1, Type 2
	Workload ordering	O_1, O_2, O_3

There are two key differences between clearing group mission tasks and the search group mission tasks. Clearing mission tasks required searching for known objects, whereas the Search mission task required searching for unknown objects. The participant was expected to experience increased cognitive and visual workload levels when searching for unknown objects, as they must evaluate whether an object is dangerous. Searching for known objects is a simpler identification process. The second difference is the procedure the participant followed when an item was found. The clearing group mission tasks required bringing the object to the robot for scanning, whereas the robot explicitly instructed the participant not to touch any objects that may be potentially dangerous or suspicious during the Search mission task. The participant was instructed to take a picture of the object, and then provide a detailed description of the object and the surrounding environment via Walkie-Talkie to one of the experimenters, who posed as the Incident Commander (see Chapter 5.1.4). Further mission task details are presented in Chapter 5.1.6.

The between-subjects variable *training session type* was randomly assigned and deter-

Table 5.2: Mission task groups and corresponding training session types.

Task Group	**Task**	**Training Session**
Clearing tasks	Pharmacy	Type 1
	Pawnshop	Type 2
Debris task	Debris	Type 1 & 2
Sampling tasks	Liquid Sampling	Type 1
	Solid Sampling	Type 2
Search tasks	Search	N/A

mined the tasks a participant performed during their training session. Training on only a subset of mission tasks results in participants encountering unknown mission tasks during the trial session, which will support the development of task recognition and workload estimation and algorithms when the tasks are unknown. There were two training session types, each consisting of one clearing group mission task, one debris group mission task, and one sampling group mission task (see Table 5.2). Specifically, Type 1 training sessions involved the Pharmacy, Debris, and Liquid Sampling tasks, and Type 2 training sessions involved the Pawnshop, Debris, and Solid Sampling tasks. No participant was trained to perform the Search task, and experienced that task for the first time during the trial session.

The missions were designed to incorporate a wide range of tasks, eliciting a broad range of workload levels across all seven components. Each mission task was capped at ten minutes and the within-subjects *task density* (i.e., workload levels) was manipulated by the number of composite/atomic tasks initiated during each mission task (e.g., number of controlled substances to clear, number of samples to collect). The workload at every mission task was elicited by increasing or decreasing the corresponding mission's composite tasks' frequency in three levels, each corresponding to a relative workload level (i.e., UL, NL, and OL). The workload ordering variable had six workload transitions (e.g., UL-NL, OL-UL), ensuring that each transition occurred exactly once per participant. Each workload level was experienced for 10 minutes, before transitioning to the next level. Three between-subjects workload orderings were used:

- O1: UL-NL-OL-UL-OL-NL-UL
- O2: NL-OL-UL-OL-NL-UL-NL
- O3: OL-UL-OL-NL-UL-NL-OL.

It is also important to note that the workload ordering and task density variables were manipulated only during the trial session. The training session had participants performing the Type 1 or Type 2 mission tasks at the NL workload level, with a five-minute break between each mission task for debriefing. Additionally, the mission tasks occurred in the same order for every participant during the trial session, and only the workload levels experienced within each mission task were varied and transitioned every ten minutes.

5.1.2.1 Secondary Tasks

Secondary tasks were included in order to introduce task concurrency, and elevate the participant's workload levels via multi-tasking. These secondary tasks are referred to as *Incident Command prompts*, and require the participant to listen for and respond to prompts over a Walkie-Talkie. Two types of Incident Command prompts were utilized. The first prompt type relayed relevant information to the participant. Example prompts include: i) "Team 10, a suspicious person has been sighted running south on Anderson Road with a black bag", and ii) "Team 10, access to Southtown is now restricted due to chemical hazards." Participants were required to acknowledge the prompts upon receiving the information and verbally relay the information to the robot teammate. Additionally, participants had to respond only to the prompts directed to their team (i.e., Team 10), and ignore prompts directed to other teams.

The participant was also asked to memorize a list of ten names during the pre-session briefing. The names represented two hypothetical task support teams that were working alongside the participant's team.

- Team 2 (Relief shelter): Kathy Johnson (Team Lead), Mark Thompson, Bill Allen, Tammy Hudson, and Matt Smith.
- Team 4 (Triage zone): Mariah Castillo (Team Lead), Liam Watson, Veronica Campbell, John Mckenzie, and Dahlia Young.

Participants were given two minutes to memorize the list just before the session began. The second prompt type involved questions incorporating these names that were posed periodically by an experimenter, who posed as the simulated disaster response scenario's Incident Commander throughout the trial. An example question was: "Team 10, can you name someone responsible for setting up the relief shelter?" The accuracy of these

responses served as a workload metric, as prior work demonstrated that secondary task accuracy decreases as workload increases [80].

Condition	Time (i.e., minute mark)									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
UL				✓					✓	
NL		✓		✓		✓		✓		✓
OL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 5.3: Timing of secondary task questions.

Workload was manipulated by increasing the secondary task frequency (see Table 5.3). The UL condition had two secondary tasks that were posed at the third and eighth minutes. Both of these tasks were relevant information relays and no questions were asked related to the memorized list of names. The NL condition had five total prompts, one every two minutes. One of the NL prompts was an irrelevant information relay, one was a relevant information relay, and two were questions about the memorized names. The OL condition had ten prompts. The first prompt was administered fifteen seconds into the condition, followed by one every minute thereafter. Two OL prompts were irrelevant information relays, three were relevant information relays, and five were questions about the memorized names. The Incident Commander also provided reminders of the remaining time to the team during the primary tasks by saying "Team 10, you have X minutes left before you need to move on to your next mission task." Additional time reminders were given over the Walkie-Talkie at the 7.5 and 9-minute marks.

5.1.3 Dependent Variables

The dependent variables included physiological metrics collected via wearable sensors, in-situ workload ratings, secondary task performance, a task timing log, a demographics questionnaire, and a post-session questionnaire. The wearable sensors were used to collect various physiological metrics. An experimenter wearing all sensors is shown in Figure 5.1, these sensors included: BioPac Bioharness BT, Xsens Mtw Awinda, an eye tracker (i.e., Pupil Labs Core and Neon), two Myo armbands, two Shure microphones (one unidirectional and one omnidirectional), and a noise meter. The sensors correspond to the activity components, as presented in Table 3.1.

The Bioharness chest-strap heart rate monitor, worn under the shirt and contacting

(a) Front View (b) Side View

Figure 5.1: An experimenter wearing all of the sensors.

the skin, produced a range of physiological metrics (e.g., heart rate, HRV, respiration rate, postural magnitude). The Xsens consisted of 17 motion trackers placed at various locations on the body, as depicted in Figure 3.2, that measured acceleration and angular velocity for different body parts (e.g., hands, feet, shoulders) [207]. The Pupil Lab's eye trackers (i.e., Core and Neon) provided ocular metrics (e.g., pupil diameter, blink rate, gaze location). A pair of Myo armbands were worn around the thickest part of the forearms [239] and generated an 8-channel sEMG signal, as well as inertial metrics (i.e., acceleration and angular velocity). The unidirectional microphone captured the participant's speech, while the omnidirectional microphone recorded the ambient environmental noise. Lastly, the noise meter provided a highly accurate measurement of the environmental noises' decibel level.

The evaluation used two different eye trackers. The Pupil Core eye tracker required using an external laptop to record and store the ocular metrics, while the Pupil Neon required an Android phone to collect the metrics. Initially, the evaluation began with the Pupil Lab's Core eye tracker; however, the eye tracker malfunctioned frequently (e.g., the laptop shutting down due to overheating and regular software crashes), and failed to

record the ocular metrics. Only two participants' sessions were successfully recorded with the Core tracker. Further investigation into the malfunction and reaching out to Pupil Labs indicated that the Core eye tracker is not suitable for recording data in a highly dynamic and mobile scenario. The Core eye tracker was replaced with the Pupil Neon, which was designed to work in such conditions, beginning with the twenty-fourth participant onward.

A laptop was used for collecting and storing the noise meter data locally. This same laptop was used to record the ocular metrics from the Pupil Labs Core wearable eye tracker. The participants who donned the Neon eye tracker had it connected to an Android mobile phone. The laptop and the Android phone were secured in a backpack, which also housed the noise meter and omnidirectional microphone. The backpack's laptop weight also simulated the additional weight of the respirators worn by civil support personnel. The backpack was secured with three buckles (i.e., shoulder, chest, and waist) to prevent sway while performing the tasks, shown in Figure 5.1.

Participants responded to the in-situ subjective workload metrics that required a participant to rate their workload levels (i.e., cognitive, visual, speech auditory, gross and fine motor, and tactile) from 1 (little to no demand) to 7 (extreme demand). The in-situ rating prompts were provided by the Incident Commander, in person, at the six-minute mark of every task. It is important to note that all secondary tasks and in-situ workload ratings were incorporated into the IMPRINT Pro models prior to the evaluation, but were not used in the experimental analyses for this dissertation.

The last dependent variable was the timing log of composite tasks with associated logged tasks. Each mission task was decomposed into composite tasks prior to the evaluation when developing the IMPRINT Pro models in order to ensure that the overall workload, as well as the individual components' workload levels, was sufficiently manipulated for each mission task. This evaluation's mission tasks allowed the participant to execute composite tasks (e.g., discovering controlled substances, and moving large boxes) as needed, which led to the composite tasks occurring at different times within a given mission. An experimenter (i.e., the Data Monitor) carefully monitored the participant's behavior and logged the exact time the majority of the composite and some sub-composite and atomic tasks occurred during a given mission.

A precise composite task timing log is necessary to associate the gathered wearable sensor data with the recorded ground truth labels, and is a robust means of aligning the IMPRINT Pro model generated results with the actual composite task execution time. Some composite tasks allow the atomic tasks to be executed in different orders. It is

important to note that the task logs only captured the composite and some sub-composite and atomic tasks (i.e., logged tasks), but not all the atomic tasks, as it was not feasible to log the atomic tasks in such a highly dynamic setting (see Section 5.1.7).

5.1.4 Experimenter Roles

This evaluation required three experimenters, referred to as the Incident Commander, the Pilot, and the Data Monitor. The Incident Commander primarily acted as the remotely located officer in charge of the operation, and broadcast secondary task messages over the Walkie-Talkie. This experimenter was located separately from the participant for the majority of the trial, but entered the environment to administer the in-situ workload ratings. The Incident Commander also served as the logistic lead and was responsible for conducting the setup procedure at the start of each session. The setup procedure included greeting the participant, administering a consent form, administering a demographics questionnaire, assisting the participant with the wearable sensors, calibrating the wearable sensors, and providing participants with their mission objectives.

All the mission tasks, except the Pharmacy task, were prepared by the three experimenters prior to the participant's arrival. The experimenters also ensured that the mission tasks were prepared to emulate appropriate workload levels depending on the workload ordering assigned to the participant. The Pharmacy task was prepared by the Pilot and Data Monitor experimenters while the Incident Commander was assisting the participant in donning and calibrating the sensors in the Setup room (see Figure 5.2) in order to avoid biasing the participant when they initially arrived at the facility.

The Pilot was responsible for monitoring the robot's behaviors, intervening when needed, executing any necessary Wizard-of-Oz behaviors relative to the robot motions, and verbal interactions. The verbal interactions were heavily automated, and followed pre-generated scripts. The Pilot simply hit the enter key, indicating to the robot to move on to the next phrase in the script. The participants were allowed to ask the robot task-relevant questions, but irrelevant questions and banter (e.g., "Hey Eve. Got any weekend plans?", "Eve, can you tell me a joke?") were ignored. Most of the task-relevant questions were anticipated and the robot had a pre-generated response (e.g., "The marker and sticky notes are located inside the cart"), which was triggered by the Pilot. Nevertheless, participants inevitably asked the robot questions for which the pre-generated responses were insufficient. The Incident Commander occasionally asked the participants to confer with

the robot, and the participant's phrasing altered how the robot needed to respond. For example, the Incident Commander asked the participant how many guns were secured at the Pawnshop. A participant may phrase the question "Eve, how many guns did we find at the Pawnshop?", to which the robot had to respond with a number. A participant may also phrase the question "Eve, we found 15 guns at the pawnshop right?", requiring the robot to respond with a yes or a no. Custom responses were primarily used to resolve these situations.

The Data Monitor was responsible for monitoring the wearable sensor feeds for any abnormalities and carefully monitoring the participants' actions, logging the tasks performed by the participants (see Chapter 5.1.7). Monitoring the physiological data streams is necessary to prevent unwanted data loss. The Data Monitor used a Python-built dashboard interface to monitor the streams for all wearable sensors, except those running on the backpack laptop. This interface was capable of notifying the experimenter when a sensor was no longer collecting data, and enabled the experimenter to restart the sensor remotely. There were rare occasions where physical intervention was required to resolve an issue, and the Data Monitor was responsible for these interventions. Precise task logging is necessary for both task recognition and workload estimation model development, as reliably associating ground truth workload values with the participant's actions is paramount to a model's performance. A custom command line terminal program was developed using Python to log the tasks' start and end times during the mission.

5.1.5 Environment and Robot Overview

The evaluation scenario simulated the aftermath of a tornado that hit a small town in Arkansas. Participants were paired with a semi-autonomous robot teammate (i.e. Wizard-of-Oz [115]) to form a disaster response team, *Team 10*. The team completed a series of seven mission tasks often performed by different first response groups (e.g., police, fire, civil support). These mission tasks include all tasks listed in Table 5.2. These mission tasks were chosen to elicit a diverse set of atomic and composite tasks that covered the breadth of the first response groups' capabilities, and used different combinations of the seven activity components. Further, the mission tasks are representative of real-world disaster response scenarios. Each mission task lasted approximately 10 minutes.

This evaluation was conducted in an off-campus warehouse facility. A layout of the warehouse is presented in Figure 5.2, which also shows the mission task locations. Each

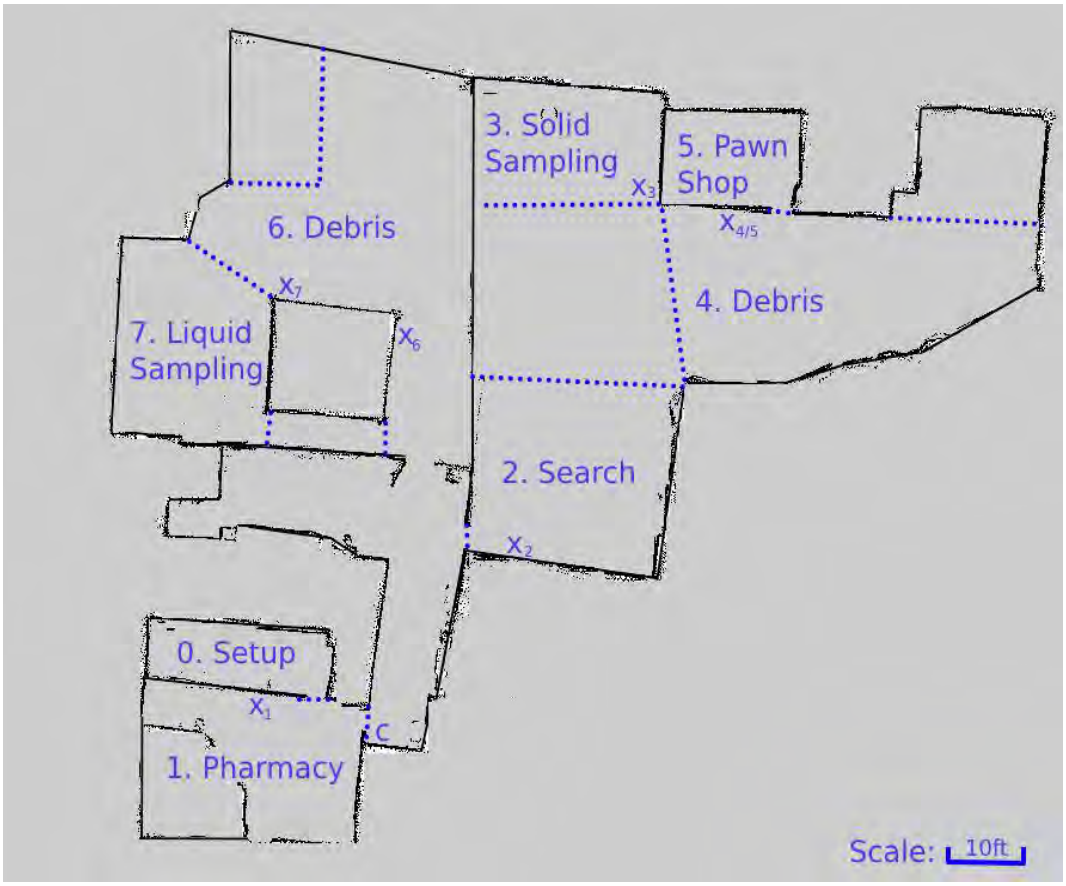


Figure 5.2: Experimental environment map generated by the Pioneer 3DX robot's LiDAR. Each task area is labeled with the mission task name and a number i representing the order in which tasks were completed during the data collection trial. The dotted lines represent boundaries delineating task areas from transition and out-of-bounds areas, where additional experimental material was stored. The front door is marked by two lines in the Pharmacy task area, and the cart's starting location is marked with a C. The markings X_i indicate the locations, where the Pilot and Data Monitor were stationed when the participants performed the mission task i.

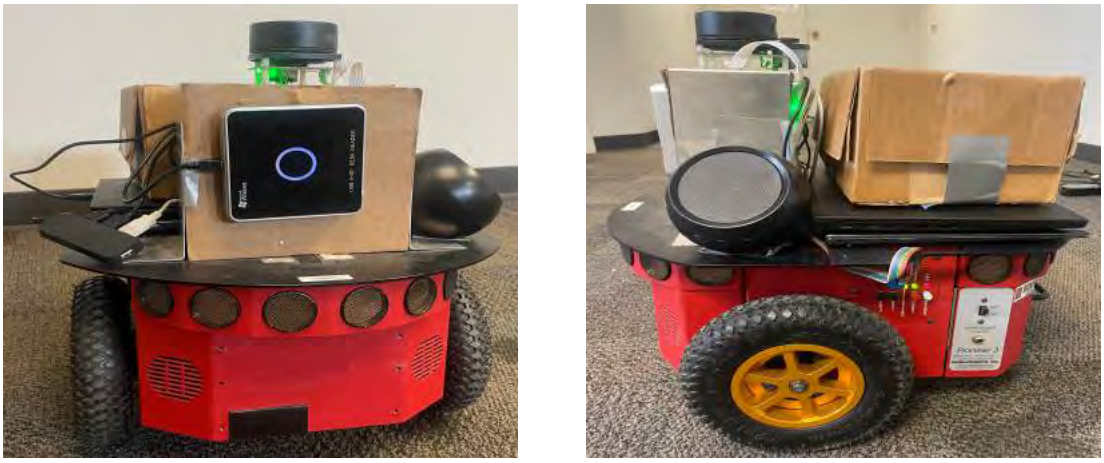
specific mission task (e.g., Pharmacy task) was always performed in the same location, for both training and trial sessions to maintain environmental consistency. Each area is labeled with a number corresponding to the order in which tasks were performed on the trial day, where there was one setup area and seven mission task areas. Further, Figure 5.2 is drawn to scale and a distance reference is located in the lower right-hand corner.

The facility's front door was located in task area one (i.e., Pharmacy task area), which was always empty of experimental material when participants arrived to avoid bias. The *Setup* area was a room in the facility's front area where all experimental logistics (e.g., putting on sensors, and filling out questionnaires) were conducted. A cart was placed near the exit of the Pharmacy task area, just outside the dotted line, and contained items the participant needed in order to perform the mission tasks. The cart location is marked with a C in Figure 5.2.

All mission task areas were located in large open spaces, except mission task areas one and five (i.e., Pharmacy and Pawnshop). Both of these clearing group mission task areas were segmented to create two spaces, so that the participant and the robot searched each space independently. The Pharmacy task area contained two rooms. The left-hand room was blocked by overturned shelves, making it inaccessible to the robot; thus, the participant searched this space. The right-hand room was more open allowing the robot to navigate freely. The Pawnshop task area contained one room, but was segmented using overturned shelves that spanned the length of the room, shown in Figure 5.3.

Figure 5.3: Pawnshop task area.

It is important to note that the transition area between the Pharmacy task and Search



(a) Front View (b) Side View

Figure 5.4: Pioneer 3DX named Eve.

task required the participant walk by two doors, one door leading to the Search task and the other leading to the final Debris task. The door leading to the Debris task was deliberately closed during the initial portions of the trial session, to avoid confusing the participant. The Incident Commander opened this door while the participant performed the Pawnshop task.

The participant was paired with a semi-autonomous robot teammate to accomplish their tasks. The Pioneed 3DX robot (see Figure 5.4) teammate was named Eve. The robot was equipped with a LiDAR, a front-facing RFID reader (Figure 5.4a), a speaker (Figure 5.4b), and a small storage box.

The Robot Operating System (ROS) was used to integrate the robot's mobility, environmental sensing, and interaction capabilities. Two computers on the robot's back served as the primary computing resources. A Linux operating system laptop was required to run ROS, which controlled the robot's motor functions via ROSARIA. The second Windows-based computer was required for the RFID reader's driver software. The RFID receiver was an 860-960 MHz UHF Gen 2 USB Plug-Play Keyboard Emulation Desktop RFID Reader from GaoRFID. GaoRFID provides a C# API for interacting with the RFID tags programmatically, which was leveraged to enable the robot to automatically detect RFID tags in the environment and then autonomously execute a corresponding appropriate behavior. An example of such behavior required the robot when a weapon RFID tag was detected

to seek assistance from the participant by saying "Hey [PARTICIPANT NAME], I think I found a weapon. Can you help me out?". A Python wrapper was developed for the C# API to leverage ROS' communication capabilities to send RFID information between the Windows and Linux laptops.

The robot used the Google Text-to-Speech Python package to verbalize all text, and had a female voice. All communication between the participant and the robot was verbal. The robot provided task-specific instructions to the participant (e.g., "place the pipette on the drop cloth"), and asked the participant for help (e.g., "Hey [PARTICIPANT NAME], I think I found a controlled substance. Can you help me out?"). The participant was instructed to speak to the robot if they had any questions or if they needed it to perform a specific duty (e.g., scan an object). The robot was programmed with a set of pre-generated answers to expected questions (e.g., "Hey Eve. Can you scan this?") and simple yes or no answers that were triggered by another experimenter, who acted as the robot's Pilot. The Pilot experimenter was co-located with the team, and had a command line terminal for triggering pre-defined responses and typing custom responses as needed. It is important to note that the pre-generated responses comprised the bulk of communication between the participant and the robot.

The LiDAR was primarily used for autonomous navigation, though this was not the robot's primary navigation mode. The Pilot remotely controlled the robot within a task, and the robot autonomously navigated to the starting locations between tasks. The robot's LiDAR was used prior to the evaluation to map the warehouse environment, so that the robot was able to navigate between task areas (shown in Figure 5.2). The Pilot monitored the robot at all times to ensure it did not crash or run into the participant. Further information on the Experimenters' roles is presented in Chapter 5.1.4.

RFID tags were placed in the environment in order to support the robot's mission task-relevant autonomous behaviors. The tags were used to represent objects (e.g., controlled substance containers, and dangerous weapons). Preset verbal responses were triggered when any RFID tag was sensed, and varied across tasks. For example, the robot notified the participant about the object, and provided them with task-specific instructions (e.g., hold the object in front of the RFID reader, take a picture of the object). Further details on mission task-specific instructions are presented in Chapter 5.1.6.

The robot also had a small storage box attached to its back into which the participant placed small items for the robot to transport. The Debris clearing task had a high volume of small items. The participant was instructed to fill the robot's box with as many small

items as possible, and notify the robot when it was full. The robot transported those items to the dumping location, where the participant emptied the box.

5.1.6 Mission Tasks

This evaluation consisted of four mission groups (see Table 5.2): clearing, debris, sampling, and search. These mission groups resulted in six unique mission tasks: 1) Pharmacy task, 2) Pawnshop task, 3) Solid Sampling task, 4) Liquid Sampling task, 5) Debris task, and 6) Search task. The materials required to complete all mission tasks are discussed, each mission task is described in detail, and a mission task decomposition into composite and associated atomic tasks is presented. It is important to note all mission tasks contained a *Null* task that accounted for the transitory interval or time periods during which the participant was not actively engaged in any composite or atomic task.

IMPRINT Pro was used to develop workload models for each decomposed mission task by workload level, prior to conducting the evaluation. IMPRINT Pro allows users to construct complex task networks, where nodes can be organized sequentially, concurrently, and hierarchically. The mission task, along with the tasks' associated composite and atomic task decomposition were used to construct the IMPRINT Pro workload model. The atomic task directly maps to IMPRINT Pro's workload anchor values; however, restricting all atomic tasks to these anchor values may misrepresent the nature of some composite tasks. Therefore, some atomic tasks' workload values had to be extrapolated from the anchor values to be sufficiently representative.

Atomic tasks were aggregated into the listed composite tasks, which were further aggregated into mission tasks. The frequency of these atomic and composite tasks was manipulated to reflect the desired task density for a given relative workload condition. It is important to note that the workload component anchor values are not normalized across components; thus, the association between a task and the allocated workload value varies significantly across components. For example, a *conversation* is anchored to a speech workload value of 4.0, while *keyboard typing* is set to a fine-grained motor value of 7.0. The mapping between IMPRINT Pro's anchor values and atomic tasks is provided in the prior Chapter (see Table 4.2).

5.1.6.1 Mission Task Materials

Mission tasks primarily had the participants interact with items in the environment; however, some composite tasks required additional materials. A cart was placed near the exit of the Pharmacy task area, just outside the dotted line on the right. This location is indicated by a blue 'C' in Figure 5.2. This cart contained all the items the participant needed to complete each mission task including a permanent marker, sticky notes, a GoPro camera, six dry sampling kits, and six wet sampling kits. Some mission task areas were separated by a reasonable distance (e.g., areas five and six). Storing mission task-specific materials in the environment was not ecologically valid, so participants pulled this cart between tasks during the training and trial sessions.

5.1.6.2 Pharmacy Task

The Pharmacy task was a clearing mission task, and required searching a collapsed pharmacy for any controlled substances. These substances came in multiple containers (e.g., loose pills in bags, unmarked bottles, and boxes). All controlled substances corresponded to one of three categories: 1) opioids, 2) stimulants, and 3) benzodiazepines.

Table 5.4: Clearing mission group's task density by workload condition.

Workload Condition	**Analysis Time**	**Number of Items**
UL	60 seconds	5 items
NL	15 seconds	10 items
OL	1 second	30 items

The participant was informed that the robot was equipped with a specialized *"scanner"* to analyze the drug type; however, the robot autonomously selected a random classification after a pre-determined wait time, shown in Table 5.4. The participant placed the discovered item in front of the robot's scanner for classification (see Figure 5.5). The robot instructed the participant to place the item in one of the three storage bins once scanning was complete, where each bin corresponded to a controlled substance category (e.g., opioids). Empty pill casings represented the controlled substances placed throughout the environment, along with empty pill bottles and boxes corresponding to uncontrolled substances (e.g., vitamins) to serve as clutter. If the participant asked the robot to scan

an uncontrolled substance container, the robot informed the participant that this item did not need to be collected and to continue their search.

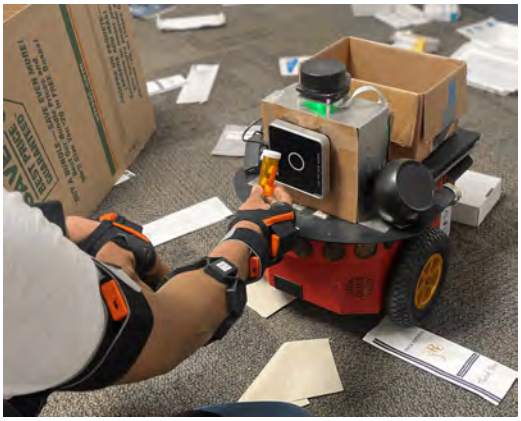


Figure 5.5: Pharmacy task, where the robot is scanning a bottle.

The participant and their robot teammate searched the environment independently, but collaborated on item analysis, classification, and storage. The participant was responsible for areas inaccessible to the robot, and the robot searched areas where it moved freely. RFID tags were placed near the controlled substances in the open areas to enable the robot to search and discover items independently. The robot verbally requested the participant's assistance upon discovering a relevant item. The participant held the item in front of the robot's scanner, and after the pre-determined wait time, the robot verbally communicated the box in which to store the item (e.g., "Stimulant. Box 2"). The task transitioned at the ninth minute when the Incident Commander asked the participant to count the number of containers in each bin, write the count for each bin on a sticky note, and place the sticky note on the bin. The task was completed when all three bins were counted and labeled.

The Pharmacy task can be decomposed into two atomic tasks and six composite subtasks: i) Null, ii) *Walking* to and from the robot, iii) *Using Walkie-Talkie* to communicate, iv) *Searching* for controlled substances, v) Waiting for the *robot to analyze* the substances, vi) *Assisting the robot* to pick up the discovered items, vii) *Dropping off items* at their respective storage bins, and viii) *Counting* all items and writing the number on a sticky note.

The robot provided instructions on how to perform the Pharmacy task during the trial for untrained participants. The robot explicitly associated the Pharmacy task with the other clearing mission task (i.e., Pawnshop task) on which the participants trained, informing them that 1) they were responsible for searching the cluttered room, 2) the robot was responsible for the open area, and 3) the analysis/classification procedure was similar.

Task density was manipulated by altering the time taken by the robot teammate to "analyze" the containers, and the total number of items that needed to be secured (see

Table 5.4). The robot took 60 seconds to analyze the item during the UL condition, and required the team to secure five containers. The robot required 15 seconds to analyze an item during the NL condition, and the team needed to secure 10 containers. Finally, the OL condition had the robot analyze an item for 1 second, and the team to secure 30 containers. Additionally, task density for all tasks included secondary tasks. The UL condition had two secondary tasks that were posed at the third and eighth minutes. The NL condition had five total prompts, one every two minutes, and the OL condition had ten prompts, one every minute. More information on these secondary tasks is provided in Chapter 5.1.2.

5.1.6.3 Pawnshop Task

The Pawnshop task was the other clearing mission task, and required searching a cluttered pawnshop to identify and collect fake guns, bullets, and grenades. Generally, this task was similar to the Pharmacy task, but was conducted in a single room. The mission task area included more inaccessible areas that the participant searched, while the robot teammate searched more open areas (depicted in Figure 5.3). The participant brought any discovered weapons to the robot to be scanned for fingerprints, then items were deposited in a storage bin corresponding to whether or not fingerprints were detected. The appropriate storage bin was verbally communicated by the robot (e.g. "Fingerprints detected, Box 1"). The scan times varied by workload condition and were the same as the Pharmacy task, as shown in Table 5.4. RFID tags were placed throughout the environment to enable the robot to search and locate items independently. The task transitioned at the ninth minute when the Incident Commander asked the participant to count the number of items with and without fingerprints, write the number of items in each bin on separate sticky notes, and place each note on their respective bins. The task was completed after both bins were labeled.

The robot provided instructions on how to perform the Pawnshop task during the trial session for the untrained participants. The robot explicitly associated the Pawnshop task with the Pharmacy task, informing the participants that 1) they were responsible for searching areas blocked by shelves, 2) the robot was responsible for the open area, and 3) the analysis/classification procedure was similar. The only differences were the dropping-off and counting pertained to weapons instead of pill containers.

5.1.6.4 Solid Sampling Task

The Solid Sampling task belongs to the sampling mission group, and required the participant to collect samples of several solid "hazardous" substances. These substances were composed of colored sand or flour, and were located in containers (e.g., clear plastic storage containers, glass jars, film canister) throughout the environment. The team was informed that the robot was sent a list of substance locations, and the robot led the participant to the first sampling location upon arriving at the task area. Samples were located on elevated surfaces (e.g., table and turned-over trash can), as well as on the floor (see Figure 5.6). The cart contained all the necessary sampling kits. Each dry sampling kit contained two sandwich-sized zip-lock plastic bags, one four-ounce glass sample jar, one stainless steel scoopula, and two alcohol wipes. The kit was wrapped in a diaper to maintain sterility and protect the kit from breakage. A permanent marker was also placed in the cart.

Figure 5.6: Solid Sampling task.

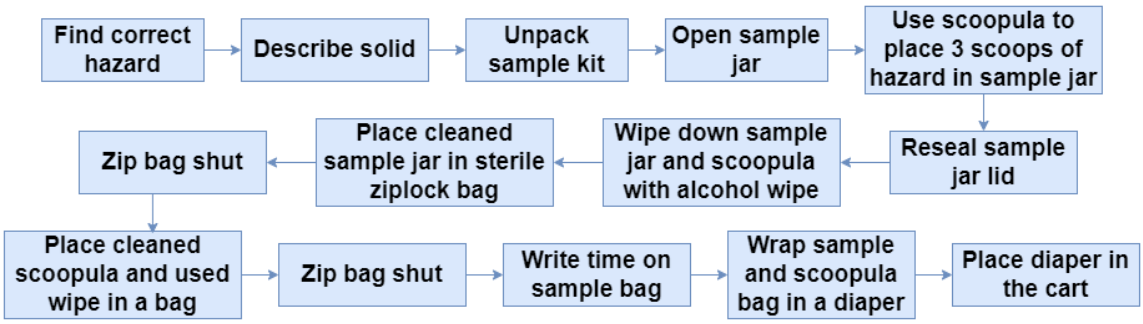


Figure 5.7: Step-by-step procedure to safely collect solid samples.

The robot asked the participant to describe the hazardous material's appearance in detail. The robot verbally confirmed that the participant's description was recorded and proceeded to provide a step-by-step procedure for how to sample the hazardous substance safely, shown in Figure 5.7. Each box in this figure represents a single instruction. The robot presented the instructions one at a time, and the participant was trained to ver-

bally confirm it was ready to move on to the next step (e.g., "Next"). This procedure followed strict guidelines for maintaining safe and sterile sampling procedures by published government standards [167].

Table 5.5: Sampling mission group's task density by workload condition.

Workload Condition	**Number of Samples Time**	**Stalling Dialog?**
UL	1 sample	Yes
NL	3 samples	No
OL	5 samples	No

The Solid Sampling task can be decomposed into one atomic task and eight composite tasks. These were: i) *Null*, ii) *Using Walkie-Talkie*, iii) *Walking with cart* between the sample locations, iv) *Describing the sample*, v) *Unpacking the kit*, which involves fetching the Solid Sampling kit from the cart and laying it next to the hazard, vi) *Sampling* the contaminant according to the robot's instructions, vii) *Writing the code* and the current time provided by the robot, viii) *Packing the kit* by wrapping the diaper with all the items and placing it in the cart, and finally ix) *Incident Commander stalling* the participant to gather information pertaining to prior tasks.

The *Incident Commander stalling* served as a means of controlling task density, while maintaining ecological validity. Generally, task density was manipulated by changing the total number of solids that required sampling (see Table 5.5). The UL condition consisted of one solid sample task, which takes approximately 3 minutes on average to complete. Prior work controlled task density by instructing participants to only begin sampling the next substance when they heard a ping [80]. Utilizing pings introduces artificial constraints on the participant's behavior, preventing them from performing their task and reducing the experiment's ecological validity. This evaluation used a dialog between the Incident Commander and the participant at the start of this task to both delay the execution of the primary task objective and to maintain realism. This dialog asked participants to report more specific information about a prior task (i.e., the Search task). The dialog also consisted of numerous pauses, as the Incident Commander often "needed to confer with another team", to ensure that workload remained in the UL range. Overall, this dialog lasted approximately 4-5 minutes depending on the level of detail provided by the participant. If a participant completed their task early, then they were instructed to hold their position and wait for further instructions. The NL and OL conditions did not require

this precollection dialog. The NL condition consisted of three samples and the OL condition consisted of five samples.

5.1.6.5 Liquid Sampling Task

The Liquid Sampling task belongs to the sampling task group, and was very similar to the Solid Sampling task. The participant's and robot's roles were identical to that of the Solid Sampling task. The primary difference was that this task required sampling "hazardous" liquids, instead of solids (see Figure 5.8). A similar highly structured protocol, based on government requirements, was used to ensure the sterile and safe collection of hazardous liquids [167]. The specific steps required for each liquid contaminant sample collection are presented in Figure 5.9.

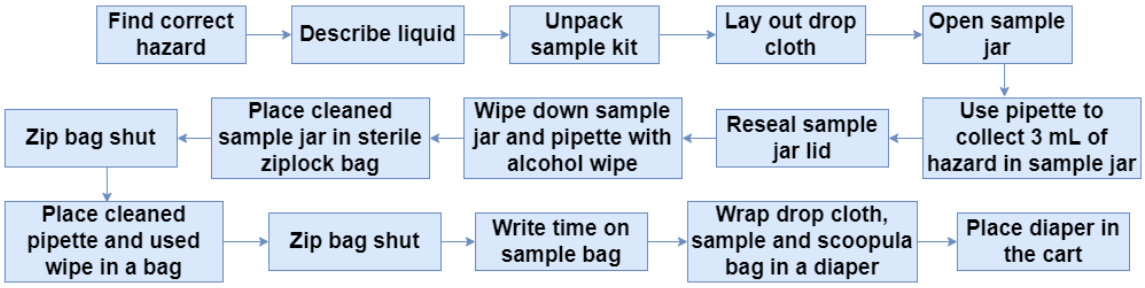


Figure 5.9: The steps completed for each liquid contaminant sample collected in the liquid contaminant sampling task.

Hazardous liquids were all different colored water, either stored in an open plastic container or spilled onto an overturned barrel. The participant was equipped with a cart that contained the required pre-assembled liquid sampling kits, labeled "wet". These kits contained two sandwich-sized zip-lock plastic bags, one four-ounce glass sample jar, one plastic pipette, one plastic drop cloth, and an alcohol wipe. The drop cloth was placed under the sampling area to catch potential spillage. The kits were wrapped in a diaper to maintain sterility and protection from breakage. Task density was manipulated in an identical manner to the Solid Sampling task (see Table 5.5). The only difference from the

solid sampling task related to task density was the UL stalling dialog pertained to the Pawnshop task, rather than the Search task. The Pawnshop task was chosen, because it was difficult to create meaningful questions about the prior task (i.e., Debris task).

5.1.6.6 Debris Task

The debris task required clearing a debris field, containing numerous large and small objects obstructing the path forward. The team moved the objects to a marked location near the debris field (see Figure 5.10). The large objects included cardboard boxes (weighing 20 to 25 lbs), chairs, and buckets with hardened cement (weighing around 30 lbs). The small objects were approximately 3 to 8-inch styrofoam blocks. The participant was trained to load the robot's box with as many small objects as possible, and then to pick up a large object as the robot transported the small objects to the dumping location. This pattern was repeated until the entire field was clear. The participant was also instructed to prioritize a path through the debris, as it may be impossible to move all the debris within the 10-minute time limit (e.g., OL condition).

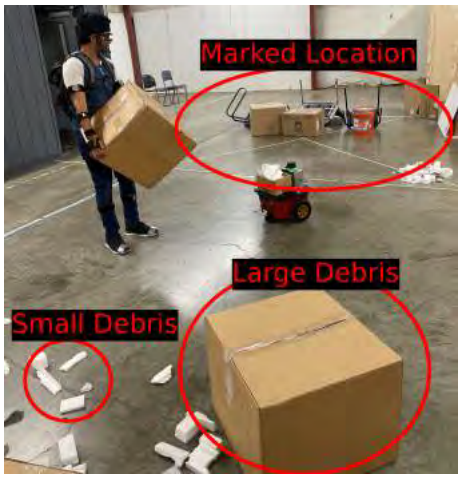


Figure 5.10: Participant moving a large object, as the robot moved small objects.

The debris task consisted of two atomic tasks and five composite tasks: i) *Null*, ii) *Walking* back the debris field, iii) *Using Walkie-Talkie*, iv) *Moving large objects* to the dumping location, v) *Loading* the robot's box with small debris, vi) *Unloading* the robot's box, and vii) the *Incident Commander stalling* the participant about a prior task. The Incident Commander's stalling dialogue pertaining to the Solid Sampling task was employed to control the UL workload levels.

The task density was manipulated by varying the total number of large objects that needed to be cleared, as well as the number of trips made by the robot (see Table 5.6). Further, there was only enough small debris to require the robot to take approximately 20 total trips. Trip numbers varied based on how full the robot's box was and if the participant elected to carry some small debris themselves. The UL condition required the

Table 5.6: Debris mission group's task density by workload condition.

Workload Condition	Number of large objects	Number of robot payloads	Stalling Dialog?
UL	10	10	Yes
NL	20	20	No
OL	40	20	No

participants to move 10 large objects, and reduced the amount of small debris even further for the robot to make approximately 10 trips. The NL condition consisted of 20 large objects, and enough small debris for the robot to make approximately 20 trips. The OL condition consisted of 40 large objects, and sufficient small debris for the robot to make 20 trips. Therefore, the OL condition shifted the majority of work onto the participant, requiring them to move two large objects for every robot transported payload.

5.1.6.7 Search Task

The Search task required the team to conduct an exhaustive inspection of an environment for potentially dangerous or suspicious items. No participants were trained to perform this task; thus, the robot provided instructions on the team's goal and responsibilities upon discovering the task. The robot explained that the team needed to deviate from its original plan, and investigate the Search task area. The robot stated that the participant was to search the area independently, and verbally communicate if a potentially dangerous item was discovered. The robot also informed the participant that they were to use the GoPro camera in the cart and instructed the participant on how to take pictures using the GoPro. When the participant found a potentially suspicious item, the robot instructed the participant to take pictures of the items.

The Search task area was an open space on the larger side of the warehouse (see Figure 5.2). A trash can, wooden crate, rolled-over plastic barrel, bulletin board, two plastic pallets, metal shelving units, and several cardboard boxes were present in the environment, shown in Figure 5.11. The red circles in this Figure represent the location of potentially dangerous items. These Dangerous items consisted of a fake pipe bomb, containers labeled "Danger. Hazardous Waste", notices on a bullet board detailing a rendezvous location, instructions for building a pipe bomb, and information on C4 explosives. These items were chosen as they are obviously dangerous or suspicious, enabling the participant to more

Figure 5.11: Search task area, where dangerous/suspicious items are outlined in red.

easily identify and evaluate them.

The team began inspecting the environment and discussed whether any discovered items were suspicious. RFID tags were placed throughout the environment enabling the robot to discover items independently; thus, either teammate was able to determine whether the team needed to investigate an item. Items that were deemed dangerous or suspicious triggered a dialog. First, the robot verbally informed the participant they were sending the item's location to Incident Command. Next, the robot asked the participant to take a picture of the object with the GoPro camera and report the item to Incident Command via the Walkie-Talkie. The Incident Commander asked follow-up questions based on the desired task density (see Table 5.7). These questions asked the participant to either evaluate the physical appearance of the object, estimate whether the object was an explosive, and assess if there were any flammable or dangerous chemicals nearby. Given the Search mission task's objectives, it can decomposed into two atomic tasks and four composite tasks: i) *Null*, ii) *Walking* through the environment, iii) *Using Walkie-Talkie*, iv) *Searching* for suspicious items, v) *Taking pictures* of suspicious items, and vi) *Describing the suspicious* item to Incident Command via the Walkie-Talkie.

The task density was controlled by manipulating the total number of items to be in-

Table 5.7: Search mission group's task density by workload condition.

Workload Condition	Number of items	Number of Questions
UL	3 items	1 question items
NL	5 items	2 questions
OL	10 items	2 question + clarification

vestigated and the number of Incident Commander follow-up questions the participant responded to upon reporting an item, shown in Table 5.7. The team was required to find three items in the UL condition, the NL condition had five items, and the OL condition had ten items. Further, the UL condition asked one follow-up question, the NL condition asked two follow-up questions, and the OL condition asked two follow-up questions and prompted the user to provide additional details if the answers were too short (i.e., <10 seconds). Additionally, secondary questions were incorporated into the tasks, and secondary task density was manipulated based on workload conditions.

5.1.7 Task Decomposition

The six mission tasks and the accompanied secondary tasks can be decomposed into a total of twenty-one composite tasks and two atomic tasks, as summarized in Table 5.8. Mission tasks within a task group (see Table 5.2) were closely associated with one another; therefore, several composite and atomic tasks were shared across multiple missions. Specifically, two atomic tasks (i.e., Null and Walking) and the Using Walkie-Talkie composite task were shared across most mission tasks, while the Incident Command stalling, and Searching composite tasks were shared across three mission tasks (see 5.8). The Null task was a placeholder to account for the transitory intervals between tasks, as well as to indicate the absence of all other composite tasks.

Each of the twenty-one composite tasks can be decomposed into sub-composite tasks and atomic tasks by incorporating different combinations of the seven activity components. An accurate composite task decomposition is required to detect the atomic and composite tasks reliably. This dissertation relies on precise knowledge of when exactly the sub-composite tasks and atomic tasks occurred within each composite task to associate the gathered wearable sensor data with the corresponding tasks to train the machine learning algorithms.

Table 5.8: The mission and secondary tasks by the corresponding atomic and composite tasks. NOTE: Grey cells represent the composite (or atomic) task's association within the corresponding mission task. The atomic tasks are highlighted in Blue.

Composite and Atomic Tasks	**Mission Tasks**					
	Pharmacy	Pawnshop	Solid Sampling	Liquid Sampling	Debris	Search
Null						
Walking						
Using Walkie-Talkie						
Incident command stalling						
Searching						
Robot analyzing						
Assisting robot						
Drop-off item						
Counting items						
Walking with cart						
Describe sample						
Unpacking kit						
Sampling						
Write sample code						
Packing kit						
Moving heavy item						
Loading						
Unloading						
Taking picture						
Describe suspicious item						
Secondary Tasks						
Incident command secondary prompt						
Incident command reminder						
In-situ ratings						

Moreover, certain composite tasks can allow for the sub-composite or the atomic tasks to be executed in different orders, or even permit omitting some sub-composite or atomic tasks, contingent on the scenario and the human teammate's current state. For example, the *Using Walkie-Talkie* composite task typically requires the participant to detach the Walkie-Talkie from their waist buckle, raise the Walkie-Talkie to the mouth, before pressing the button to speak; however, it is possible to press the button to speak prior to detaching the Walkie-Talkie, or press the button to speak without detaching the Walkie-Talkie at all. Therefore, accurately labeling the low-level atomic tasks during their execution becomes more complex in such a highly dynamic environment.

An intermediate task state, *logged task*, and a task decomposition based on task logging is presented. Logged tasks refer to tasks that are recorded reliably and systematically. These logs typically include the execution time, duration length, and task description. Logged tasks represent a compromise between the theoretically possible hierarchical task decomposition and the practicality needed to label decomposed tasks during the evaluation. Each of the twenty-one composite tasks was decomposed coarsely into its logged task constituents by aggregating the atomic tasks that were not labeled individually.

- The *Using Walkie-Talkie* composite task was composed of three logged task constituents: i) a fine-grained motor and tactile component of *reaching out* and *grasping* the Walkie-Talkie, ii) a tactile *pressing* and holding the Walkie-Talkie, and iii) conversing with the Incident Command that encompassed an auditory *Incident Command communication* component, a cognitive *conversation* component, and v) speech *Incident Command information* response component.

- The *Incident Command stalling* task's decomposition was similar to the *Using Walkie-Talkie* task, but had cognitive *memory recall* element instead of the conversation aspect.

- The *Searching* composite task was composed of two logged task constituents: i) searching constituent that encompassed gross motor *walking*, visual *scanning* the environment, and cognitive *evaluating* the items, and ii) picking up items that included a gross motor *bending* over, fine-grained motor *picking* up items during the Pharmacy and Pawnshop missions, as well as clearing the wooden crate and trash can during the Search mission task, and tactile *grasping* items.

- The *Waiting for the robot to analyze* composite task entailed four logged constituents:

i) gross motor *bending* or *squatting*, ii) tactile *holding* of the item, iii) speech *scan request* to the robot, and iv) auditory *analyzing item* prompt from the robot.

- *Assisting the robot* composite task's decomposition was identical to the Searching task, but also included an auditory *assist request* from the robot.
- *Drop-off item* incorporated three logged constituents: i) visual *locate* to identify the box, ii) gross motor *walking* and *bending* over, and iii) tactile *holding* item.
- *Counting the items to write on sticky notes* consisted of four constituents: i) visually *locating* the items, ii) cognitive *counting* of items, iii) fine-grained motor and tactile *writing*, and iv) auditory request from the robot to report to the Incident Command after writing the count.
- The *Walking with cart* composite task had only one logged constituent, which was a combination of gross motor *walking* and tactile *holding* the cart.
- *Describing the sample substance* was composed of three logged constituents: i) auditory *description request* from the robot, ii) examining the sample that encompassed visual *inspection* and cognitive *evaluation*, and iii) providing speech *sample description* to the robot.
- The *Unpacking and packing the sampling kits* tasks were composite twins in that they entailed identical constituents, but differed in the order in which the constituents occurred. These two composite tasks comprised: i) fine-grained motor and tactile *packaging*, ii) visually *inspecting* the contents, and iii) cognitive *association*.
- The *Sampling* composite task consisted of two logged constituents: i) listening auditory *sampling instructions* from the robot and cognitive *processing* of the information, and ii) visual *coordination*, as well as fine-grained motor and tactile *sampling* to gather the substance.
- *Writing the sampling code and time* involved two logged constituents: i) auditory *sample code* and time information and the accompanied cognitive *processing* of the information, and ii) fine-grained motor and tactile *writing*.
- *Moving heavy item* task encompassed: i) tactile *lifting* heavy object, and ii) gross motor *carrying* heavy object and *walking*.

- *Loading the robot's box* contained four logged tasks: i) visual *locating* small debris, ii) gross motor *bending* over, iii) fine-grained motor *picking* up items to deposit in the robot's box, and iv) tactile *grasping* the small debris items.
- *Unloading the robot's box* task's decomposition was identical to the loading task, but did not include the visual component.
- The *Taking pictures* composite task comprised: i) fine-grained motor *reaching out* for GoPro, ii) tactile *pressing* the shutter button, and iii) cognitive *association*.
- *Describing a suspicious item* encompassed four logged constituents tasks: i) examining the suspicious item that encompassed visual *inspection* and cognitive *evaluation*, ii) a fine-grained motor and tactile component of *reaching out* and *grasping* the Walkie-Talkie, iii) a tactile *pressing* and holding the Walkie-Talkie to communicate with the Incident Commander, and iv) speech *suspicious item information* to the Incident Commander.
- The *Incident Command secondary prompt* included: i) auditory *secondary prompt* from the Incident Commander, ii) cognitive *memory recall*, and iii) speech *prompt response* either to the robot or the Commander.
- The *Incident Command reminder* had one logged constituent that included an auditory *time reminder* from the Commander, and cognitive *processing* of the remaining time information.
- The *In-situ ratings* task comprised: i) auditory *in-situ probe* from the Commander, ii) cognitive *evaluating* the workload levels, and ii) speech *in-situ response* to the Commander.

5.1.7.1 Summary

Several logged tasks were shared across multiple composite tasks; however, the order in which the logged tasks occurred and their duration distinguished the composite tasks from one another. The individual task recognition algorithms developed in Chapter 3.5 were evaluated to recognize the logged tasks for each task relevant activity component. This intermediate level was the lowest hierarchical level for which reliable ground truth task

Table 5.9: The logged tasks identified for each activity component across all composite tasks. The *Null* task associated with each activity component indicates an absence of the other tasks.

Activity Component	**Logged tasks**
Gross motor	Walk, Bend over, Carry large object, Null
Fine-grained motor	Package, Pick, Reach, Sample, Write, Null
Tactile	Grasp, Hold, Lift, Package, Press, Sample, Write, Null
Visual	Coordination, Inspect, Locate, Scan, Null
Cognitive	Association, Conversation, Count, Evaluation, Process, Recall, Null
Auditory	Robot's analyze prompt, Robot's assist request, Robot's sample description request, Robot's sampling instructions, Robot's request to report to Incident Command, Incident Command communication, Incident Command reminder, In-situ probe, Secondary prompt, Null
Speech	Sample description, Suspicious item information, Incident Command information, Scan request, Secondary response, In-situ response, Null

labeling during the evaluation (i.e., task execution time and duration) was obtained. It is worth noting that these individual algorithms can detect the actual atomic tasks without loss of generality, provided the atomic tasks' reliable ground truth data is available. The logged tasks identified for each activity component (see Table 5.9) across the twenty-one composite tasks can be summarized as follows:

- The Gross motor activity component had a total of four logged tasks: i) *Walking*, ii) *Bending over* to pick the items, or to hold the items in front of the robot, iii) *Carrying large object* while clearing debris, and iv) *Null*.

- The fine-grained motor activity component contained six logged tasks: i) *Packaging* the kits, ii) *Picking* up items, iii) *Reaching* out for Walkie-Talkie and GoPro, iv) *Sampling* the contaminants, v) *Writing*, and vi) *Null*.

- The tactile activity components included eight logged tasks: i) *Grasping* items, ii) *Holding* items and cart, iii) *Lifting* heavy objects, iv) *Packaging* the kits, v) *Pressing* the Walkie-Talkie or GoPro, vi) *Sampling* the contaminants, vii) *Writing*, and viii) *Null*.

- The visual activity component entailed five logged tasks: i) *Coordination* while sampling the contaminant, ii) *Inspecting* the contaminants, or suspicious items, iii) *Lo-*

cating known items (e.g., pills and weapons in the storage bins, small debris) iv) *Scanning* the environment for pills, weapons, or suspicious items, and v) *Null*.

- The cognitive activity component consisted of seven logged tasks: i) a cognitive *Association* task for less mental workload tasks (e.g., taking a picture, unpacking, and packing sampling kits), ii) *Conversation* when interacting verbally with the robot or Incident Command, iii) *Counting* the number of pills and weapons, iv) *Evaluation* for assessing contaminants and suspicious items, v) *Recalling* information from memory when responding to Incident Command's interrupts and secondary prompts, vi) *Processing information* provided by the robot or Incident Command, and finally, vii) *Null*

- The auditory activity component comprised ten logged tasks: i) the robot's prompt for *analyzing items*, ii) the robot's *assistance request*, iii) the robot's *sample's description request*, iv) the robot's *sampling instructions*, v) the robot's request to *report to the Incident Commander* after a task was completed, vi) *Incident Commander's reminder*, vii) *Incident Commander's communication*, viii) *Secondary prompt* from the Incident Commander, ix) *In-situ probes*, and x) Null.

- The speech activity component included seven logged tasks: i) providing *Sample description* to the robot, ii) *Suspicious item description* and providing contextual information to the Incident Commander, iii) *Information* to the Incident Commander, iv) *Scan request* to the robot, v)*Secondary prompt response*, vi) *in-situ response*, and vii) *Null*

5.1.8 Procedure

This evaluation was conducted over two days. The first day consisted of a 30-minute training session to familiarize the participants with the robot, and help them understand the robot's role. The second day consisted of the full 70-minute trial session.

5.1.8.1 Training Session

Upon arrival, the Incident Commander greeted the participant, introduced the participant to the other experimenters, and began explaining the session. First, the participant was informed of the session's duration (i.e., 2 hours), the pariticipant's financial compensation,

and that they were allowed to stop at any time for any reason. The participant read and signed the consent form, followed by completing a demographic questionnaire. The Incident Commander then began assisting the participant with donning the wearable sensors, which included: BioPac Bioharness BT, Xsens Mtw Awinda, Pupil eye tracker (i.e., Core or Neon), two Myo armbands, two Shure microphones (one unidirectional and one omnidirectional), and a noise meter.

The sensor donning process began with the Incident Commander demonstrating how to wear the Bioharness BT using a set of posters mounted to the wall, after which the Incident Commander left the Setup room with the blinds closed to allow the participant to put on the sensor. The Incident Commander entered the room and verified verbally that the sensor was put on properly. Two Xsens foot inertial trackers were duct-taped onto the participant's shoes. Xsens sensors for the calves, thighs, and waist Xsens sensors were secured using the provided velcro straps. The chest and shoulder Xsens sensors were tapped onto the participant's clothes using fabric-friendly paper tape. The participant put on a backpack and secured it tightly using the chest and waist buckles. The participant donned the two Myo sEMG sensors on the forearms that were calibrated prior to putting on each arm's bicep and wrist Xsens sensors. The Xsens hand sensors were tucked into a pair of Xsens gloves that the participant put on. The participant donned a uni-directional Shure microphone headset to capture their speech responses, while the omnidirectional Shure microphone was attached to the backpack to gather the ambient noise. The participant wore the Pupil Labs eye tracker (either Core or Neon), followed by the Xsens headband that housed the head inertial sensor.

The Incident Commander calibrated the Xsens and the two microphones, before connecting the eye tracker to its associated data collection device (i.e., Core was connected to a laptop, while the Neon was connected to an Android phone). The Core eye tracker was calibrated using the laptop, while the Neon did not require any calibration. The noise meter was then connected to the laptop and started its data collection process. Finally, the laptop along with the noise meter (and Android Phone) were secured in the backpack. The Data Monitor verified the sensor calibration and began collecting the Bioharness, Myo, Xsens, and microphone data wirelessly on the dashboard laptop. Additionally, a Walkie-Talkie was strapped onto the backpack's waist buckle to facilitate the participant's communication with the Incident Commander.

Once the sensors were calibrated, the Incident Commander began explaining the training session. The participant was informed that they were training to be a member of a

human-robot disaster response team (Team 10) who were responding to a natural disaster (e.g., tornado). The participant was informed that they were going to be working alongside a fully autonomous robot teammate. The Incident Commander explained that the participant was responsible for responding to messages over the Walkie-Talkie. The participant was tasked with ignoring messages not pertaining to Team 10, repeating informative messages directed at Team 10 to the robot, and answering questions Incident Command posed to their team (see Chapter 5.1.2).

The participant was informed that in-situ workload ratings questions were going to be asked during the training sessions. The concept of workload, as well as each individual component's contribution toward overall workload, was explained in detail. Any questions the participant had about these concepts were answered by the Incident Commander. Lastly, participants were given two minutes to memorize a list of names, and were instructed that the Incident Commander will be asking questions about this list during the session (see Chapter 5.1.2). If the participant was randomly assigned a Type 1 training session, they exited the Setup room, where the Incident Commander began explaining the responsibilities for both the robot and the participant for the Pharmacy task. If the participant was randomly assigned a Type 2 training session, they were escorted to the door of the Pawnshop (see Figure 5.2).

The participant was instructed to ask the Pilot and Data Monitor if they had any questions during the task, as these two experimenters were co-located with the participant at all times (refer Figure 5.2). The Pilot and Data Monitor provided feedback during the training task if the participant engaged in any obviously incorrect behavior (i.e., unsafe sampling practices).

The participant was trained on each assigned mission task for ten minutes. All training tasks were conducted at the normal workload condition level. The in-situ workload ratings were verbally administered at the six-minute mark, which allowed the participant to learn the meaning of the components' definitions, and become familiar with answering questions during the mission tasks. Participants debriefed with experimenters after each mission task to verify that participants fully understood their responsibilities, to allow the experimenters to provide feedback, and to answer any questions participants had. Participants immediately proceeded to the next training mission task area after this debriefing period, and the Incident Commander began explaining the roles and responsibilities for the next mission task. The training session concluded after all three mission tasks were completed. Participants who were assigned Type 1 training performed the Pharmacy mission task,

followed by the Debris mission task (marked as number 6 in Figure 5.2) and the Liquid Sampling task, while Type 2 participants were first trained on the Pawnshop mission task, followed by the Debris mission task (marked as number 4 in Figure 5.2) and the Solid sampling mission task.

The Incident Commander verbally confirmed if the participant felt sufficiently trained on how to conduct all three mission tasks with their robot partner. No explicit competency test was conducted. The Pilot and Data Monitor observed the participants and assessed their ability to perform the mission tasks. Generally, a single run through each mission task was sufficient for the participant to achieve proficiency. Corrections rarely needed to be made, and feedback from the Pilot and Data Monitor was sufficient to curb all incorrect mission task execution behaviors. If a participant felt they needed more training on a particular task, then another run through that task was offered. It is important to note that no participant was trained to perform the Search task, as it was the sole completely untrained task during the trial session.

Lastly, the participant was escorted to the Setup room, and all wearable sensors were removed in the reverse order of how they were donned, starting with the headband and eye tracker. A final post-session questionnaire was administered and the participant received their $20 financial compensation.

5.1.8.2 Trial Session

The trial session occurred at least two days after, and typically, within one week of the training session; however, four participants had scheduling complications and were unable to return for over two weeks. These participants were asked to verbally explain each of the tasks they were trained to perform. If the explanation was insufficient, the Pilot and Data Monitor demonstrated how to perform that task. Each trial session was a total of two hours long, where the experiment required 70 minutes. The trial session was composed of seven tasks presented in the same order for all participants, but with varied workload levels, where each individual task required 10 minutes. The participant was randomly assigned a workload ordering, which determined the relative workload condition for each task. Mission tasks were completed in the same location, as shown in Figure 5.2.

The participant completed the demographic questionnaire upon arrival, after which they donned the same set of sensors and equipment in the same order as the training day. The demographics questionnaire was re-administered to gather the relevant caffeine intake

and sleep-related information. The participant was briefed on their mission objective, which involved responding to a tornado that hit a town in Arkansas. The roles of the human teammate, robot teammate, and the Incident Commander remained the same. The Pilot and the Data Monitor were only tasked with operating the robot and logging the tasks, respectively, and refrained from providing feedback or assisting the participant at any point during the trial session. The participant and the robot began the trial in the Pharmacy task area and proceeded in the order shown in Figure 5.2.

Participants who completed the mission task before the 10-minute time limit were instructed to report via the Walkie-Talkie to the Incident Commander by the robot. If there was less than one minute remaining, the Incident Commander instructed the participant to follow the robot to the next task. If there was more than one minute left, the Incident Commander instructed the participant to wait. After which, the Incident Commander instructed the participant to follow the robot to their next task once the 10-minute limit was reached. Participants that were still performing the task when the time limit was reached were asked to stop what they were doing immediately, and follow the robot to the next task. The Pilot and the Data Monitor always followed the participant in close proximity (10-15 ft) when transitioning to the next mission task to maintain sensor connectivity with the dashboard laptop. The Pilot and Data Monitor remained stationed at the vantage points, marked as $\mathbf{X_i}$ in Figure 5.2, upon reaching the next mission task area.

The robot interacted with the participant upon arrival at the Search task area, and informed them the team needed to deviate from their plan. The robot asked the participant to report this deviation to the Incident Commander, and then provided the participant with the team's goal and the participant's responsibilities. Further details on this interaction and the Search mission task are provided in Chapter 5.1.6.7. The participant and their robot teammate proceeded to perform the remaining mission tasks. Any questions that the participants had regarding a mission task were answered by the robot.

The participant was interrupted at times with the secondary tasks in order to introduce concurrency, and to serve a secondary workload metric. Further details on these secondary tasks are provided in Chapter 5.1.2. In-situ workload ratings were verbally administered six minutes into the trial and every ten minutes after the initial rating. The trial concluded at the seventieth minute, or when the team finished sampling all the liquid contaminants, whichever occurred earlier. The Pilot and Data Monitor verified that the sensor data was saved, and the Incident Commander entered the environment to escort the participant back to the Setup area to remove all the wearable sensors. The participant completed the

post-session questionnaire, and then was presented with their $40 financial compensation.

5.1.9 Participants

Thirty-six participants (18 male, 17 female, and 1 non-binary) completed the full experiment (i.e., training and trial sessions). Participants were screened to ensure that they were at least 18 years old, no more than 65 years old, did not have a pacemaker or permanent metal installation in the chest area, were not pregnant, did not wear glasses, and did not have trouble lifting 35 lbs. Participants who completed both the training and trial sessions received a total of $60 compensation.

The mean age was 28.78 (std. dev. = 10.82), with a range from 18 to 60. Thirteen participants held a high school degree, ten held an undergraduate degree, eight held a master's degree, and five held a doctorate. Participants indicated the number of hours they use a desktop or laptop per week, as computer experience may impact task performance. The majority of participants (twenty-four) indicated that they use computers for more than eight hours per week. Participants also indicated if they received training on any of the following: life-guard, first-responder, civil support, or paramedic, as this experience can impact their knowledge for performing some tasks. Twenty-eight participants had no prior training in these fields, while eight participants did have prior training. Fifteen participants did not have any caffeinated drinks the day of the experiment, while fifteen participants drank at most 16 oz., three participants drank between 17 and 32 oz, and one participant drank more than 32 oz. Participants were also asked how many hours they exercise per week. Participants exercised on average 4.81 (std. dev. = 2.91) hours a week. Participants slept an average of 7.45 (std. dev. = 1.04) hours the night before the experiment and an average of 7.35 (std. dev. = 1.03) hours two nights prior. Participants rated their current fatigue levels using a Likert scale from 1 (little to no) to 9 (extreme) as 2.91 on average (std. dev. = 1.36).

5.2 Results

The task recognition algorithms were validated using the *leave-one-subject-out* cross-validation scheme, where the average accuracy is reported by training the algorithm repeatedly on all, but one participant's data and validated using the left-out participant's data [81]. Each task component algorithm is analyzed for multiple window sizes in order

to determine the window size's impact on the respective algorithms' performance.

The 70-minute trial sessions were composed of seven 10-minute mission tasks, where the participants switched from one task to the other without any break in between missions. This continued mission execution resulted in a long-tailed distribution across components, where a few tasks account for majority of the data and all other tasks are under-represented. A good example of a long-tail distribution was the Pharmacy mission's fine-grained motor tasks, where participants were in constant motion, regularly picking items from the floor, but only wrote the count of different items at the very end of the mission task; therefore, the *picking items* task account for most of the Pharmacy's fine-grained motor dataset (i.e., majority task), while the *writing* task instances were few (i.e., minority task). Training machine learning algorithms on such imbalanced datasets can bias the algorithm; therefore, the majority logged tasks' instances were randomly downsampled across components in order to ensure that the individual task recognition algorithms' accuracies were not artificially inflated due to dataset imbalance.

A random averaging downsampling method was employed, where the average number of task instances within each component, excluding the *Null* task, was calculated across all mission tasks for each participant. If a participant's count for a specific task was higher than the calculated average (i.e., one of the majority tasks), the count for that task was randomly downsampled to the average. Conversely, if a participant's count for a particular task was lower than the average (i.e., one of the minority tasks), the count for that task was not downsampled. The *Null* task's count was excluded from calculating the average across all components to prevent inflating the average count artificially for each participant. This procedure was iterated for all the logged tasks within each component, and the task instances before and after the downsampling process are detailed in the respective component's results section below.

It is important to note that the downsampling process applied solely to the individual task recognition algorithms and not to the GNN fusion and TCN composite and concurrent task recognition algorithms. This distinction arises, because the latter two are multi-label classification algorithms (i.e., predicting more than one class) that take as input the individual algorithms' task predictions or the atomic task series across components. Consequently, downsampling tasks from any single component necessitates downsampling the entire atomic task series or task predictions across all components, which is impractical. Therefore, the downsampling process was not implemented for the GNN fusion and TCN composite and concurrent task recognition algorithms.

The individual task recognition algorithms' classification accuracy was the primary dependent variable for assessing the performance, while the confusion matrices compared the individual task recognition algorithms' accuracies and misclassifications by tasks (see Chapter 4.2). Cohen's d measured the effect sizes. The Friedman's analysis of variance by ranks test was used to determine statistical significance in accuracies between results. Significant results were further analyzed using the Wilcoxon signed-rank test to identify the specific significant differences. The non-parametric statistical tests ensured that the outcomes were unaffected by the accuracy distribution across participants.

5.2.1 Cognitive Task Recognition

The cognitive task recognition algorithm incorporated thirteen features extracted from HRV, pupil dilations (left and right eyes), and blink metrics (see Chapter 3.5.1). The features were fed into a RF classifier that was trained to predict one of the seven cognitive tasks: i) Association, ii) Conversation, iii) Count, iv) Evaluate, v) Process, vi) Recall, vii) Null (described in Chapter 5.1.7). The cognitive tasks' data distribution before and after the downsampling process is presented in Table 5.10. The evaluated window sizes $t_w = \{5s, 10s, 15s, 30s, 60s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance. It is also important to note that the analysis presented is only based on the twelve participants for whom the eye tracker data was gathered (see Chapter 5.1.3). Additionally, seven of those twelve participants did not have any instances of the Association cognitive task for the 60s window size; therefore, the 60s window size is excluded from the analysis.

Table 5.10: The mean (std. dev.) and the cumulative task instances for the cognitive component before and after downsampling, aggregated across participants.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Association	329.20 (112.06)	3292	312.60 (88.18)	3126
Conversation	231.10 (51.55)	2311	231.10 (51.55)	2311
Count	79.90 (26.50)	799	79.90 (26.50)	799
Evaluate	1057.30 (226.62)	10573	459.50 (83.85)	4595
Process	557.60 (230.21)	5576	422.00 (90.71)	4220
Recall	513.80 (218.09)	5138	400.70 (112.57)	4007
Null	2363.00 (493.92)	23630	459.50 (83.85)	4595

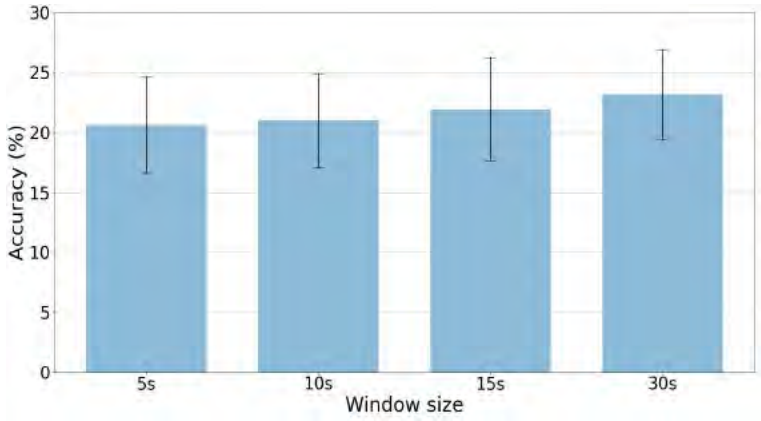


Figure 5.12: Cognitive task recognition accuracy % mean (std. dev.) by window size using the HRV, pupil dilation, and blink metrics.

The cognitive task recognition algorithm's accuracy increased with the window size, as depicted in Figure 5.12. The algorithm's accuracy for the 5s window size was 20.61% and increased gradually up to 23.16% for the 30s window size; however, the Friedman's test identified no significant differences in accuracies between the window sizes ($\chi^2(3, 11) = 3.36, p = 0.33$).

The evaluated RF algorithm's confusion matrices for the four incorporated window sizes were analyzed to identify the recognition rates by tasks (see Figure 5.13). The algorithm was heavily biased toward predicting the Evaluate, Process, Recall, and Null tasks, resulting in higher misclassification rates for the Association, Conversation, and Count tasks across window sizes. The algorithm's bias can be attributed to the RF algorithm not learning anything particularly useful from the input features and predicting tasks purely based on how the data was distributed. The long-tailed data distribution caused the Evaluate, Process, Recall, and Null tasks' instances to account for most of the cognitive data, as they occurred often. Comparatively, the Association, Conversation, and Count tasks' instances were lower, given their sporadic occurrences. For example, Evaluate was the most prominent cognitive task throughout the Pharmacy and Pawnshop missions, while the Count task only occurred at the very end of these missions.

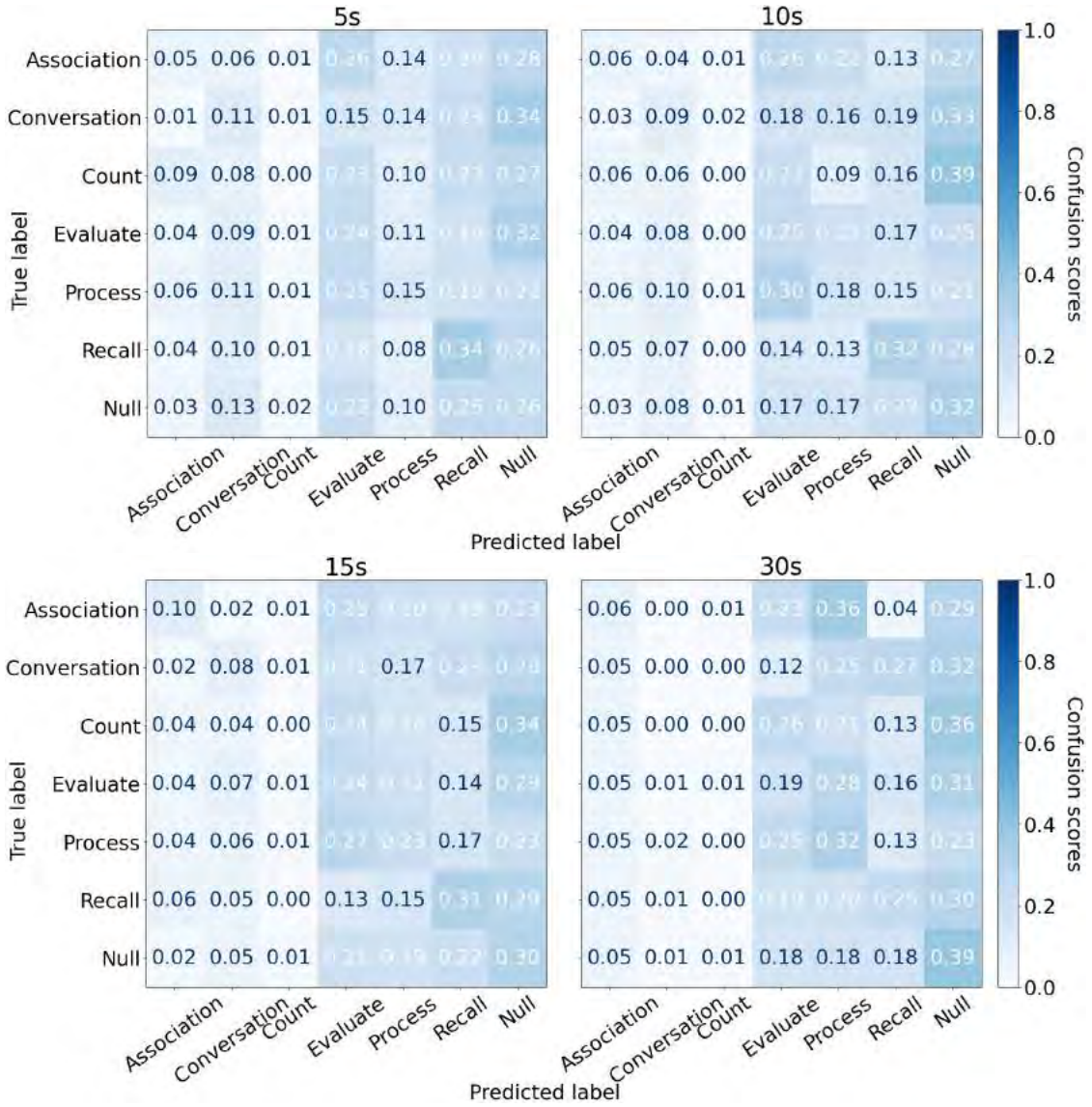


Figure 5.13: Cognitive task recognition confusion matrices for the incorporated window sizes.

5.2.1.1 Discussion

Hypothesis $\mathbf{H_1^C}$ predicted that the RF algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The algorithm's accuracy when predicting the tasks, regardless of the window size, was only 5% to 10% better than randomly guessing the tasks. The algorithm's poor classification performance was primarily due to the low-data regime, as the required metrics were available from only twelve participants. The low-data regime was further perpetuated by the long-tail distribution, causing the RF algorithm to be biased toward certain tasks.

None of the evaluated window sizes are suitable for detecting cognitive tasks in a peer-based environment. A viable alternative is to infer cognitive tasks indirectly based on the other component task detections via the GNN-based fusion algorithm.

5.2.2 Speech Task Recognition

The speech-reliant task detection algorithm incorporated the MFCCs' mean and std. dev. and the five features extracted from the speech-based metrics. The features were fed into a deep learning algorithm to predict seven tasks: i) Information to the Incident Commander, ii) In-situ response, iii) Suspicious item description, iv) Sample description, v) Scan request, vi) Secondary response, and vii) Null (described in Chapter 5.1.7). The speech tasks' data distribution before and after the downsampling process is presented in Table 5.11. Five window sizes ($t_w = \{1s, 3s, 5s, 10s, 15s\}$) with a 50% were used to evaluate the impact of the window size on algorithm's performance; however, the 10s and 15s window size had no instances for the Scan request and Sample description tasks, as participants spoke for < 10 seconds for these tasks. Therefore, the 10s and 15s window sizes are excluded from the analysis.

The algorithm's accuracy increased and peaked for the 3s window size (47.19%) and dropped at the 5s window size (43.64%) (see Figure 5.14). The Friedman's test indicated a significant accuracy difference between window sizes ($\chi^2(4, 14) = 15.86$, $p < 0.01$). The Wilcoxon signed-rank test found that the 3s window size's accuracy was significantly higher than all other window sizes with a large effect size ($p < 0.01$, $0.77 <$ Cohen's $d < 1.80$), while the 1s window size's accuracy was significantly lower than the rest ($p < 0.01$, $0.77 <$ Cohen's $d < 1.80$). Other accuracy differences were not significant.

The 3s window size's confusion matrix, depicted in Figure 5.15, was analyzed to under-

Table 5.11: The mean (std. dev.) and the cumulative task instances for the speech component before and after downsampling, aggregated across participants.

Tasks	Before Downsampling		After Downsampling	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Request robot to scan	152.94 (28.10)	5353	152.94 (28.10)	5353
Describe sample to robot	336.69 (113.96)	11784	335.74 (111.99)	11751
Provide information to IC	1690.26 (676.01)	59159	699.94 (104.43)	24498
Describe suspicious item to IC	434.43 (203.53)	15205	433.77 (202.39)	15182
Response to secondary prompt	1058.03 (300.55)	37031	679.86 (108.11)	23795
Response to in-situ probe	535.63 (141.38)	18747	516.74 (120.57)	18086
Null	12679.40 (1264.13)	443779	699.94 (104.43)	24498

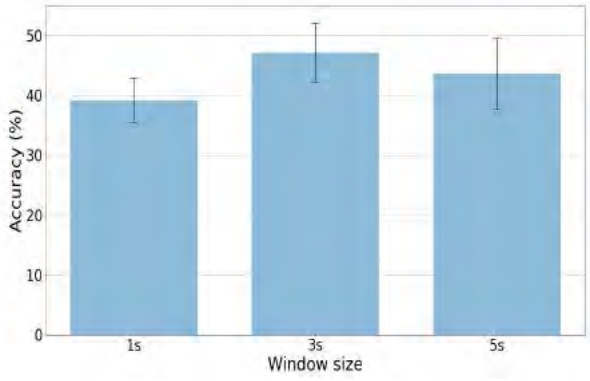


Figure 5.14: Speech-reliant task recognition accuracy % mean (std. dev.) by window size using the MFCCs and speech-based metrics.

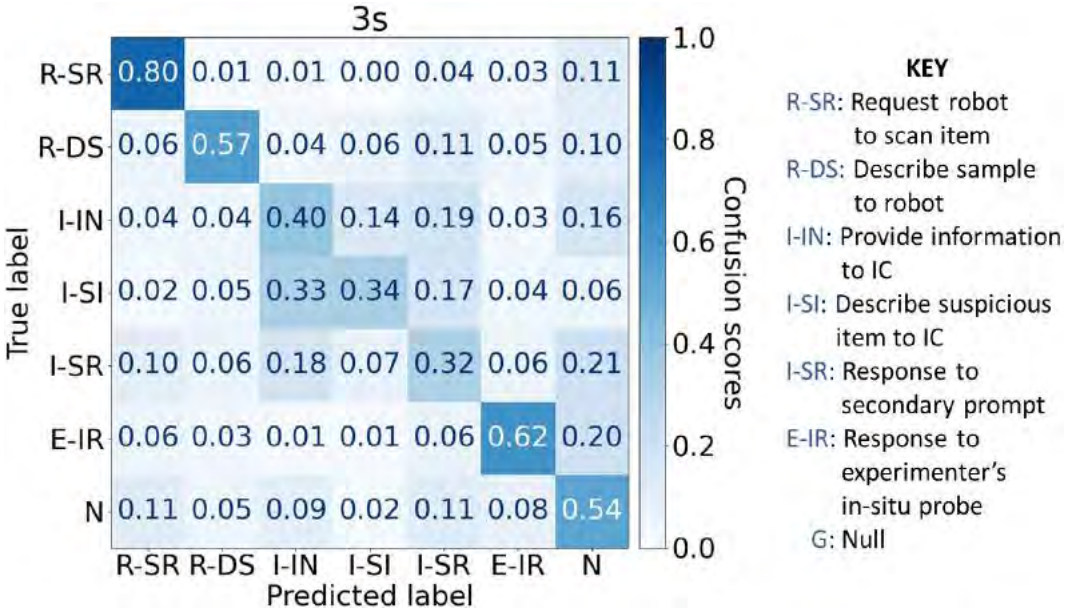


Figure 5.15: Speech-reliant task recognition confusion matrix for the 3s window size. NOTE: *IC* refers to *Incident Commander*.

stand the confusion and misclassification rate between tasks. Other window sizes' confusion matrices are provided in Appendix B Figure B.1, as the 3s window size's performance was significantly better with a large effect size. The algorithm had the highest accuracy (80%) by task for the Scan request prompt, and the lowest (32%) for the Secondary response prompts. The confusion matrix indicated that the verbal interactions (i.e., Scan request and Sample description) pertaining to the robot were detected with $\geq 50\%$ accuracy, while a majority of the speech-based interactions with the Incident Commander (i.e., Information to the Incident Commander, Suspicious item description, and Secondary response) were detected with $\leq 40\%$ accuracy. The Secondary response, Suspicious item description, and Information to the Incident Commander had the highest number of confusions among the tasks. The Secondary response task was often confused with the Null (21%) and the Information to the Incident Commander (18%) tasks, while the Suspicious item description was often misclassified as Information to the Incident Commander task (33%). The confusion between the Secondary response vs. Null tasks can be attributed to participants neglecting to answer the secondary prompts at times, while confusion among the Secondary response, Information to the Incident Commander, and the Suspicious item description tasks can be

attributed to inter-task similarities that were further exacerbated by the common mode of communication (i.e., over Walkie-Talkie).

5.2.2.1 Discussion

Hypothesis $\mathbf{H_1^S}$ predicted that the algorithm will detect the speech-reliant tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The algorithm had a severe drop in performance between the two evaluations. The supervisory evaluation (see Chapter 4.2.2) only had two speech-reliant tasks (i.e., COMM verbal response and Null), which can be basically reduced to detecting speech vs. mute. The peer-based evaluation increased the number of tasks to include events containing both complex (almost an entire sentence) and simple speech (≤ 3 words). The increased number of tasks, and inter-task similarities led to the algorithm's subpar classification performance.

Both evaluations indicated that larger window sizes are detrimental to speech-reliant task detection, as they can completely mask simple speech events, especially in an uncertain, dynamic environment, where human teammates may communicate in cryptic phrases with fewer words. The 3s window size is the recommended window size, as it demonstrated the ability to detect both complex and simple speech events with varying lengths, tones, and syllables across evaluations.

5.2.3 Auditory Task Recognition

The auditory task recognition algorithm combined the Mel spectrogram metrics obtained from an omnidirectional microphone with the time-based noise level features from a decibel meter to train a deep learning network. The algorithm predicted ten auditory tasks: i) the robot's analyze prompt, ii) the robot's assist request, iii) the robot's sample description request, iv) Incident Command communication, v) Incident Command reminder, vi) In-situ probe, vii) the robot's prompt to report to the Incident Commander, viii) robot's sampling instructions, ix) Incident Commander's secondary prompt, and x) Null. The auditory tasks' data distribution before and after the downsampling process is presented in Table 5.12. The evaluated window sizes $t_w = \{1s, 3s, 5s, 10s, 15s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance.

The algorithm's accuracy increased gradually and peaked at the 5s window size (47.96%), before it dropped at the 10s (44.17%) and 15s (31.59%) window sizes (see Figure

Table 5.12: The mean (std. dev.) and the cumulative task instances for the auditory component before and after downsampling, aggregated across participants. NOTE: IC refers to Incident Commander.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Robot's analyze prompt	289.42 (108.72)	10419	289.42 (108.72)	10419
Robot's assist request	359.00 (79.60)	12924	359.00 (79.60)	12924
Robot's sample description request	245.75 (93.36)	8847	245.75 (93.36)	8847
Robot's report to IC prompt	309.58 (75.67)	11145	309.58 (75.67)	11145
Robot's sampling instructions	1274.75 (421.45)	45891	762.36 (65.91)	27445
IC communication	2174.58 (888.94)	78285	764.11 (70.73)	27508
IC reminder	498.31 (119.83)	17939	497.42 (117.86)	17907
IC's secondary prompt	1165.47 (338.48)	41957	747.58 (71.94)	26913
Experimenter's in-situ probe	593.11 (154.73)	21352	582.25 (141.99)	20961
Null	11611.06 (935.29)	417998	765.50 (68.30)	27558

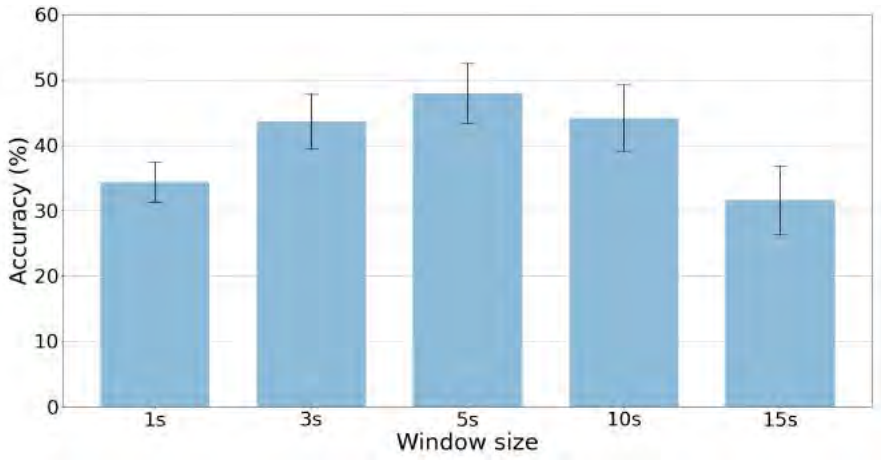


Figure 5.16: Auditory task recognition accuracy by window size when incorporating the spectrogram and noise level metrics.

5.16). The Friedman's test indicated a significant accuracy difference between window sizes ($\chi^2(4, 17) = 55.48$, $p < 0.01$). The Wilcoxon signed-rank test found that the 5s window size's accuracy was significantly higher than all other window sizes with a large effect size ($p < 0.01$, $0.74 <$ Cohen's $d < 3.64$), while the 15s window size's accuracy was significantly lower than the rest ($p < 0.01$, $1.01 <$ Cohen's $d < 3.64$). The 3s and 10s window sizes' accuracies were significantly higher than the 1s and 15s window sizes ($p < 0.01$, $2.16 <$ Cohen's $d < 2.62$). No other differences were significant.

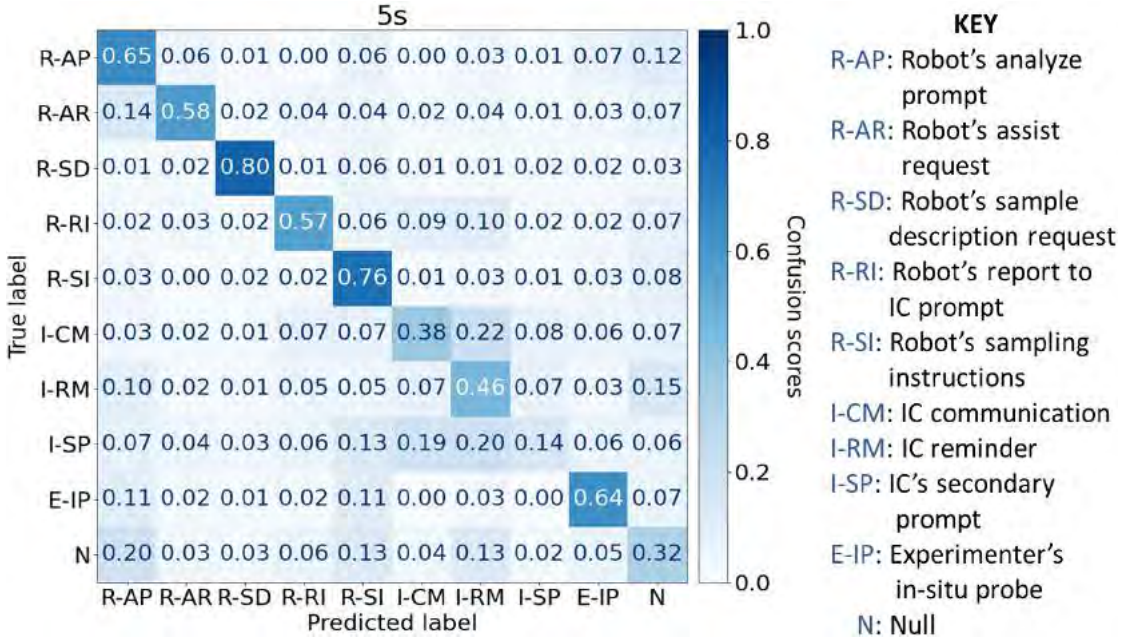


Figure 5.17: Auditory task recognition confusion matrix for the 5s window size. NOTE: *IC* refers to *Incident Commander*.

The 5s window size's confusion matrix, depicted in Figure 5.17, was analyzed to understand the confusion and misclassification rate between tasks. Other window sizes' confusion matrices are available in Appendix B Figure B.2, as the 5s window size clearly outperformed the rest. The algorithm had the highest task accuracy (80%) for the robot's sample description request prompt event, and the lowest (14%) for the Incident Commander's secondary prompts. The confusion matrix indicated that all five of the robot's prompts (i.e., robot's sample description, sampling instructions, prompt to analyze item, assistance request, and prompt to report to the Incident Commander) were detected with $\geq 50\%$ accuracy, with two tasks exceeding $\geq 75\%$. The algorithm had a high misclassification rate when detect-

ing the Incident Commander's auditory tasks (i.e., Incident Commander's communication requests, reminders, in-situ probes, and secondary prompts). The Incident Commander's secondary prompts, communication, and Null events' had the highest number of confusions among the tasks. The secondary prompts were often misclassified as the Incident Commander's communication or reminders, while the Incident Commander's task was often confused with the reminders.

5.2.3.1 Discussion

Hypothesis $\mathbf{H_1^A}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was only supported for the Robot's sample description request task. The algorithm's performance was underwhelming even when incorporating the spectrogram metrics in addition to the noise level metrics.

The supervisory evaluation only had three tasks (i.e., Radio message, Ping, and Null), but each had distinct sound characteristics, which were recognized with $\sim$80% accuracy only using the noise level metric (see Chapter 4.2.3). The peer-based evaluation increased the number of tasks to include events with varying durations; however, all of these events were purely conversational, either from the robot or the Incident Commander. The high similarity between these conversational events led to the algorithm's subpar classification performance. For example, the Incident Commander's communications, reminders, and secondary prompts, started with the same phrase *"Incident Command to Team 10, . . . "*. Additionally, intermittent auditory tasks (e.g., in-situ probes and secondary prompts) sometimes overlapped with the robot's prompts. For example, secondary prompts often coincided when the robot was providing sampling instructions during the liquid and solid sampling missions.

Auditory task detection algorithms must have the ability to identify acoustic changes within a short span, especially in a highly dynamic setting, so that the event can be detected before it switches; therefore, smaller window sizes are preferred. 5s is the recommended window size to detect auditory tasks, as it had one of the highest accuracies across evaluations. The 10s window size performed the best for the supervisory; however, the algorithm was not trained with the spectrogram metrics for the supervisory evaluation, and had a limited number of tasks, in terms of variety and duration. The 5s window size choice is also supported by literature in that algorithms that incorporate the spectrogram metric perform well with smaller window sizes [82, 144, 159].

Auditory tasks are independent of the human teammate, as the existence of ambient noise characteristics is identified from sounds in audio recordings. The developed algorithm assumed that only one auditory task occurred at any given instant; however, two or more auditory tasks may occur together (i.e., secondary prompts overlapping with the robot's sampling instructions). Thus, a polyphonic auditory detection algorithm may be required [31, 32, 204].

5.2.4 Visual Task Recognition

The visual task recognition algorithm incorporated features extracted from the eye tracker's fixations and saccades metrics, as well as the Xsens' head motion tracker's inertial metrics. The features were fed into a RF classifier that was trained to predict one of the five visual tasks: i) Coordination, ii) Inspect, iii) Locate, iv) Scan, and v) Null (described in Chapter 5.1.7) for each window. The visual tasks' data distribution before and after the downsampling process is presented in Table 5.13. The evaluated window sizes $t_w = \{5s,\ 10s,\ 15s,\ 30s,\ 60s\}$ with a 50% overlap inform the impact of the window size on the algorithm's performance. It is important to note that the analysis presented is only based on the twelve participants for whom the eye tracker data was available due to the mentioned issues with the eye tracker system (see Chapter 5.1.3).

Table 5.13: The mean (std. dev.) and the cumulative task instances for the visual component before and after downsampling, aggregated across participants.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Coordination	558.25 (148.69)	6699	558.25 (148.69)	6699
Inspect	715.58 (129.75)	8587	712.33 (121.52)	8548
Locate	1477.83 (112.94)	17734	906.58 (77.68)	10879
Scan	884.17 (101.12)	10610	846.08 (82.35)	10153
Null	2736.25 (214.52)	32835	906.58 (77.68)	10879

The visual algorithm's accuracy did not increase with the window size, as depicted in Figure 5.18. The algorithm's accuracy for the 5s window size was 43.42% and remained relatively the same across window sizes. The Friedman's test identified no significant differences in accuracies between the window sizes ($\chi^2(4, 12) = 2.87, p = 0.58$).

The incorporated window sizes' confusion matrices were analyzed to identify the best-performing window size (see Figure 5.19). The algorithm had the highest classification rate

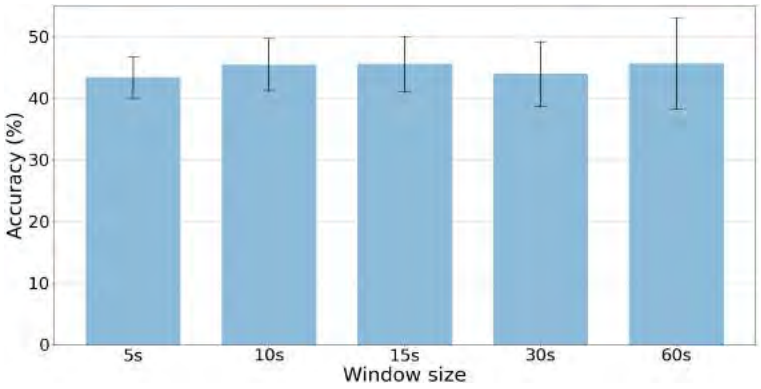


Figure 5.18: Visual task recognition accuracy % mean (std. dev.) by window size using the fixation, saccades, and inertial metrics.

for the Coordination and Scan tasks ($\geq$ 50%), while the other tasks' classification rates were between $30-40\%$ across window sizes. The Coordination task was often confused with the Inspect task and vice-versa, while the Locate and Scan tasks were often misclassified as each other. Additionally, the Inspect, Locate, and Scan tasks were frequently confused with the Null task, and vice-versa. The Coordination and Inspect tasks' classification rates increased with window size, while the Locate and Scan tasks' classification rates decreased with window size.

The window sizes' performances were highly polarized in that the smaller window sizes (i.e., the 5s and 10s) had higher misclassification rates for the Coordination and Inspect tasks, but lower misclassification rates for the Locate and Scan tasks. Contrarily, the larger window sizes (i.e., the 30s and 60s) had lower misclassification rates for the Coordination and Inspect tasks, but higher misclassification rates for the Locate and Scan tasks. The 15s window size provided an optimal classification trade-off across tasks.

5.2.4.1 Discussion

Hypothesis $\mathbf{H_1^V}$ predicted that the RF algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The algorithm's subpar classification performance was primarily due to the low-data regime, as the required visual metrics were available from only twelve participants. The polarizing performance differences between window sizes indicated that a universal window size to reliably detect visual peer-based tasks does not exist and that the window size required to assimilate the

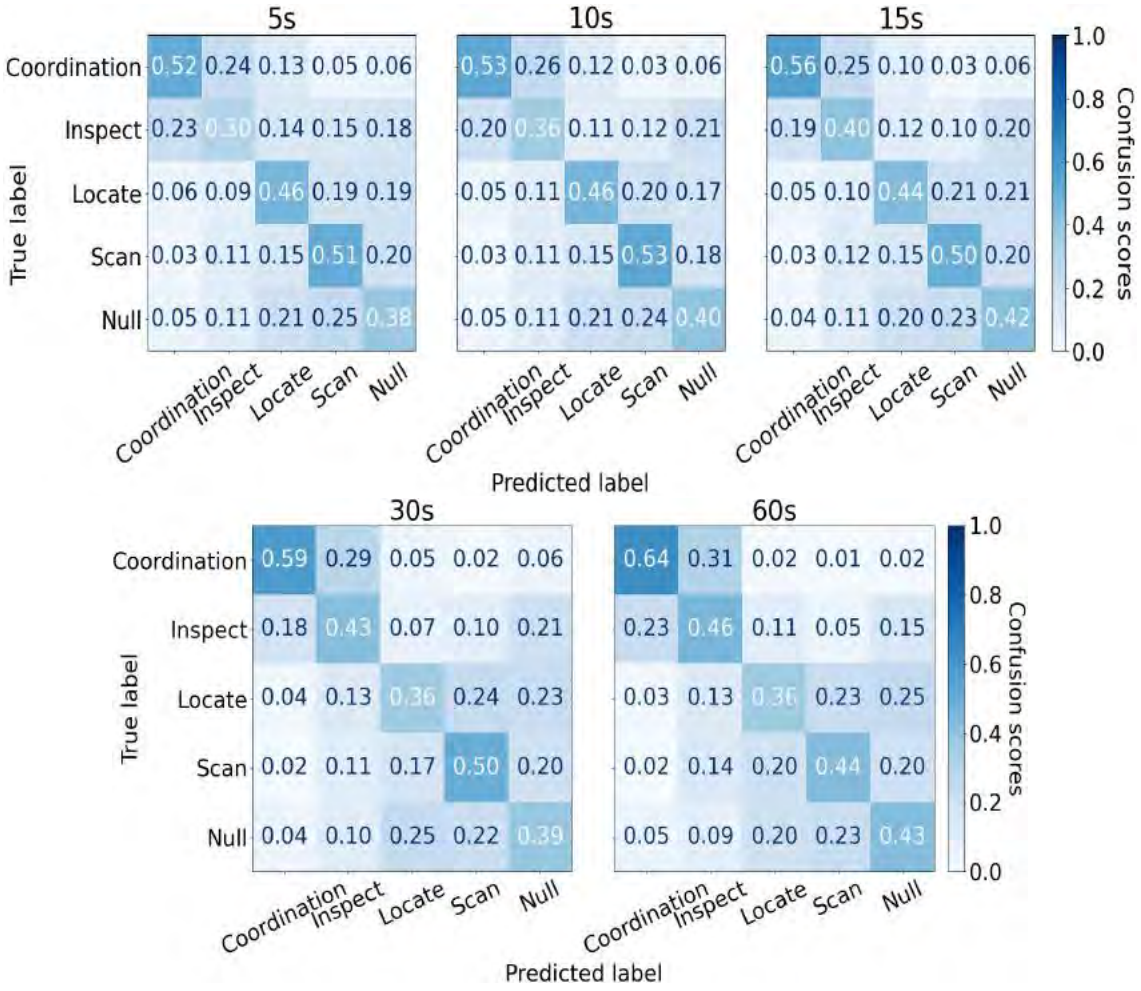


Figure 5.19: Visual task recognition confusion matrices for the incorporated window sizes.

context necessary to detect a task largely depends on the tasks' duration. Among the analyzed window sizes, the 15s is the recommended window size, as it presents a balanced performance across tasks; however, given the uncertain and dynamic nature of peer-based scenarios, visual tasks will have varied durations. Therefore, an adaptive sliding window or an ensemble window prediction method may be required to detect visual tasks.

5.2.5 Gross Motor Task Recognition

The gross motor task recognition algorithm incorporated the Xsens' pelvis, shoulders, biceps, calves, and feet IMU metrics, as well as the Bioharness' physiological (i.e., heart rate, respiration rate, and posture magnitude) metrics. The algorithm predicted four gross motor tasks: i) Bend, ii) Carry, iii) Walk, and iv) Null. The gross motor tasks' data distribution before and after the downsampling process is presented in Table 5.14. The Xsens' thigh IMU metrics were replaced with the bicep and shoulder IMUs, as the prior evaluation (Chapter 4.2.5) indicated that incorporating more than two lower-body IMU metrics was unnecessarily redundant. The bicep and shoulder IMUs were incorporated to capture the upper-body tasks (i.e., Bend and Carry). Window sizes, $t_w = \{1s, 2s, 3s, 5s, 10s\}$, with a 50% overlap, were investigated for analyzing the window size's impact on the algorithm's performance.

Table 5.14: The mean (std. dev.) and the cumulative task instances for the gross motor component before and after downsampling, aggregated across all missions and participants.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Bend	4500.24 (1403.90)	153008	3144.59 (920.93)	106916
Carry	985.38 (369.64)	33503	985.38 (369.64)	33503
Walk	3953.74 (1256.84)	134427	3144.59 (920.93)	106916
Null	7155.29 (1865.16)	243280	3144.59 (920.93)	106916

Overall, the algorithm's accuracy increased until the 5s window size (59.68%) and decreased to 58.01% for the 10s window size when incorporating the physiological and IMU metrics (see Figure 5.20). The Friedman's test indicated a significant accuracy difference between window sizes ($\chi^2(4, 36) = 35.98$, $p < 0.01$). The Wilcoxon signed-rank test found that the 1s window size's accuracy was significantly lower than all other window sizes ($p < 0.01$, $0.23 <$ Cohen's $d < 0.47$). The 5s window size's accuracy was significantly

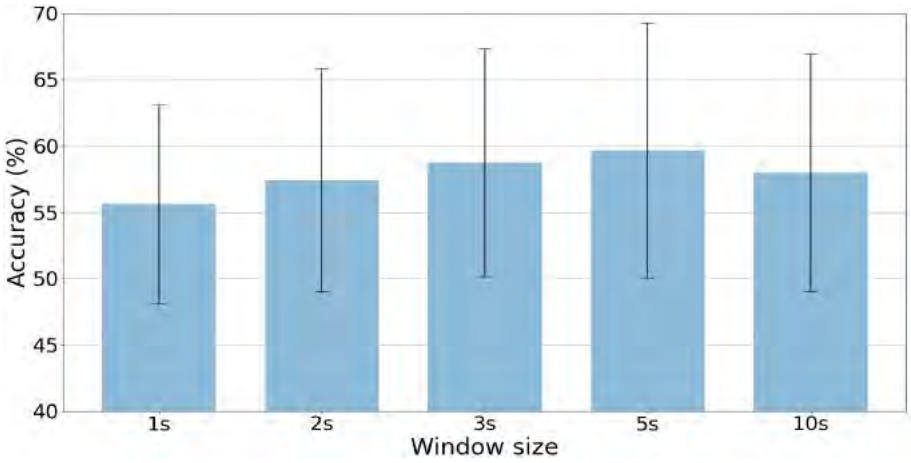


Figure 5.20: Gross motor task recognition accuracy by window size when incorporating the physiological and the IMU metrics on both limbs, shoulders, and pelvis.

higher than the 2s and 10s window sizes with a small effect size ($p < 0.01$, $0.18 <$ Cohen's $d < 0.25$), while the 3s window size's accuracy was significantly higher than the 2s ($p < 0.01$, Cohen's $d = 0.16$). No other differences were significant.

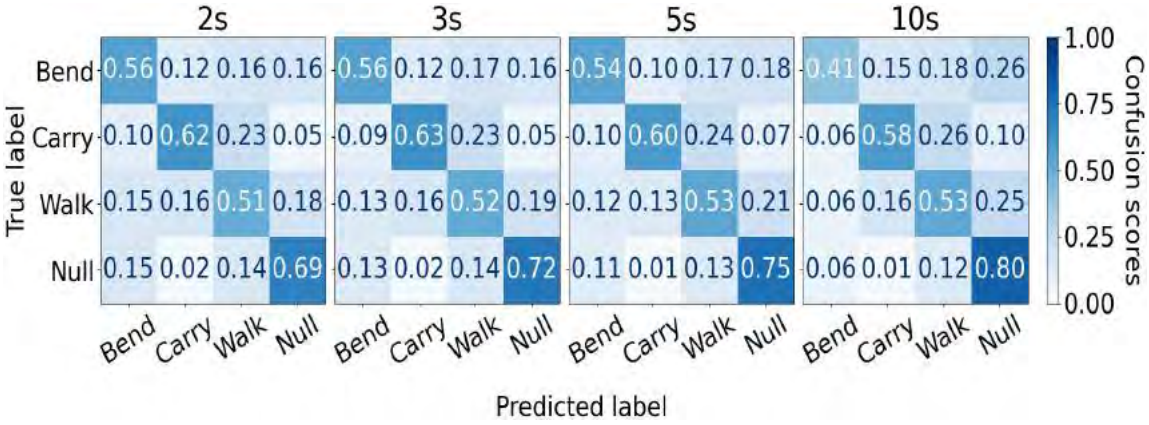


Figure 5.21: Gross motor task recognition confusion matrices when incorporating the physiological and IMU metrics for the 2s, 3s, 5s, and 10s window sizes.

The 2s, 3s, 5s, and 10s window sizes' confusion matrices were analyzed to identify the best-performing window size (see Figure 5.21). The 1s window size's confusion matrix appears in Appendix B Figure B.3, as it was significantly lower than the rest. The confusion matrices indicated that the 10s window size variant had high sensitivity for the Null task,

but had a high misclassification rate for the Bend task. The three smaller window sizes (i.e., 2s, 3s, and 5s) had similar accuracies by tasks with each variant recognizing all four tasks with $\geq 50\%$ accuracy. Among the three window sizes, the 3s had the most balanced accuracy across tasks. The 2s window size had a slightly higher confusion rate for the Null task, whereas the 5s window size had a higher confusion rate for the Bend and Carry task than the 3s window size. Generally, the Bend, Carry, and Null tasks were often misclassified as Walk, while the Walk task was often confused with the Null task. For example, the 3s window size's confusion matrix indicates that the Bend, Carry, and Null tasks were confused with the Walk for 17%, 23%, and 14% of the time, respectively, while the Walk task was wrongly misclassified as Null for 19% of the time. Similar observations can be made for other window sizes as well.

5.2.5.1 Discussion

Hypothesis $\mathbf{H_1^{GM}}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one window size, which was not supported. The algorithm's subpar performance can be attributed to three factors: low-data regime, inter-task similarity, and intra-task variability. Increasing the task variety (from two to four), while simultaneously decreasing the participant count (from sixty to thirty-six) caused a low-data regime that negatively impacted the algorithm's ability to detect the tasks accurately.

The included tasks suffered from inter-task similarity, where the tasks shared similar motion patterns. For example, the Carry task's motion patterns overlapped with the Walk task considerably, causing increased confusion between the two tasks. A similar observation can be made for the Null task, where the participants were supposed to do nothing, but moved occasionally.

Individual differences among participants resulted in the same task being performed differently. For example, participants were informed to lift the heavy object when performing the Carry task; however, there were instances where participants pushed, slid, and at times threw the heavy objects when clearing the debris, to keep up with the time pressure. This intra-task variability further exacerbated the confusion rate between tasks.

The 3s window size performed the best across both supervisory (see Chapter 4.2.5) and peer-based evaluations. Therefore, it appears to be the optimal window size for detecting gross motor tasks for the intended domain, as it offers the best trade-off between a long enough window that extracts features representative of the tasks, and short enough to

identify the logged tasks before task switching.

5.2.6 Fine-Grained Motor Task Recognition

The fine-grained motor task recognition algorithm incorporated the Xsens IMU on the hands and wrists of both arms, as well as the two Myos' forearm IMU and the 8-channel sEMG. The algorithm employed eight CNNs, where each network extracted features pertaining to each metric's left and right arms. The CNN features were combined to predict one of the six fine-grained motor tasks: i) Package, ii) Pick, iii) Reach, iv) Sample, v) Write, and vi) Null for each window. The fine-grained motor tasks' data distribution before and after the downsampling process is presented in Table 5.15. Five window sizes ($t_w = \{1s$, $2s$, $3s$, $5s$, $10s\}$) were investigated.

Table 5.15: The mean (std. dev.) and the cumulative task instances for the fine-grained motor component before and after downsampling, aggregated across participants.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Package	922.47 (405.30)	31364	922.47 (405.30)	31364
Pick	4072.91 (1348.68)	138479	1775.35 (517.13)	60362
Reach	1714.09 (558.55)	58279	1520.09 (457.28)	51683
Sample	1455.88 (589.51)	49500	1426.94 (562.74)	48516
Write	721.94 (262.97)	24546	721.94 (262.97)	24546
Null	7709.94 (2166.63)	262138	1775.35 (517.13)	60362

The fine-grained motor algorithm's accuracy did not increase with the window size when incorporating all four metrics for both arms, as depicted in Figure 5.22. The algorithm's accuracy for the 1s window size was 37.65% and remained relatively the same across window sizes. The Friedman's test identified no significant differences in accuracies between the window sizes ($\chi^2(4, 25) = 8.42, p = 0.08$).

The incorporated window sizes' confusion matrices were analyzed to understand the misclassification rate between tasks and the window sizes' impact on the tasks' accuracies (see Figure 5.23). Overall, the algorithm had the highest classification rate ($\geq 60\%$) for the Reach task, and was the lowest for Write ($\leq 21\%$) and Null tasks ($\leq 15\%$) across all window sizes. The Sample and Package tasks were often confused with one other, as these two tasks were neighboring tasks, occurring one after the other. The Write task was frequently confused with all the other tasks (excluding the Null task), because most tasks

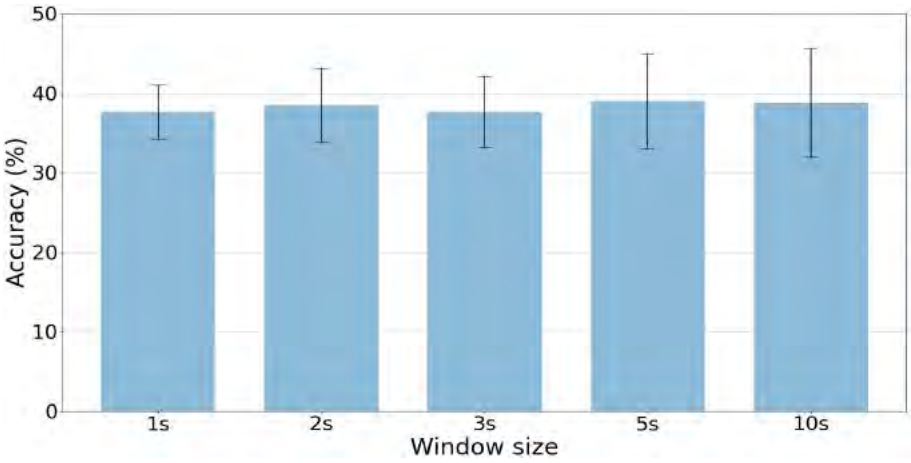


Figure 5.22: Fine-grained motor task recognition accuracy % mean (std. dev.) by window size with all four metrics from both arms.

either precede or follow the Write task. For example, the Write task is preceded by the Pick task and followed by the Reach task during the Clearing missions, while it is preceded by the Sample and followed by the Package tasks during the Sampling missions. It is also important to note that some tasks are highly sensitive to changes in window size. For example, the Sample tasks' classification rate increased with the increase in window size, while the Package tasks' rate was progressively worse. Another observation is that the Reach task's classification rate increased from 60% to 81% until the 5s window size and dropped to 70%. Overall, the 3s and 5s window sizes had higher classification rates and lower confusion across tasks when compared to the rest of the window sizes.

5.2.6.1 Discussion

Hypothesis $\mathbf{H_1^{FM}}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The significant drop in the algorithm's accuracy for detecting the peer-based tasks is two-fold. Firstly, the volume of the dataset did not increase proportionately to the increase in the number of fine-grained motor tasks detected. Additionally, the task downsampling procedure to avoid algorithm bias further reduced the data points available per task. These two attributes effectively created a low-data regime, resulting in poor algorithm optimization. Lastly, most misclassifications occurred during transitions between tasks, implying that inertial-based

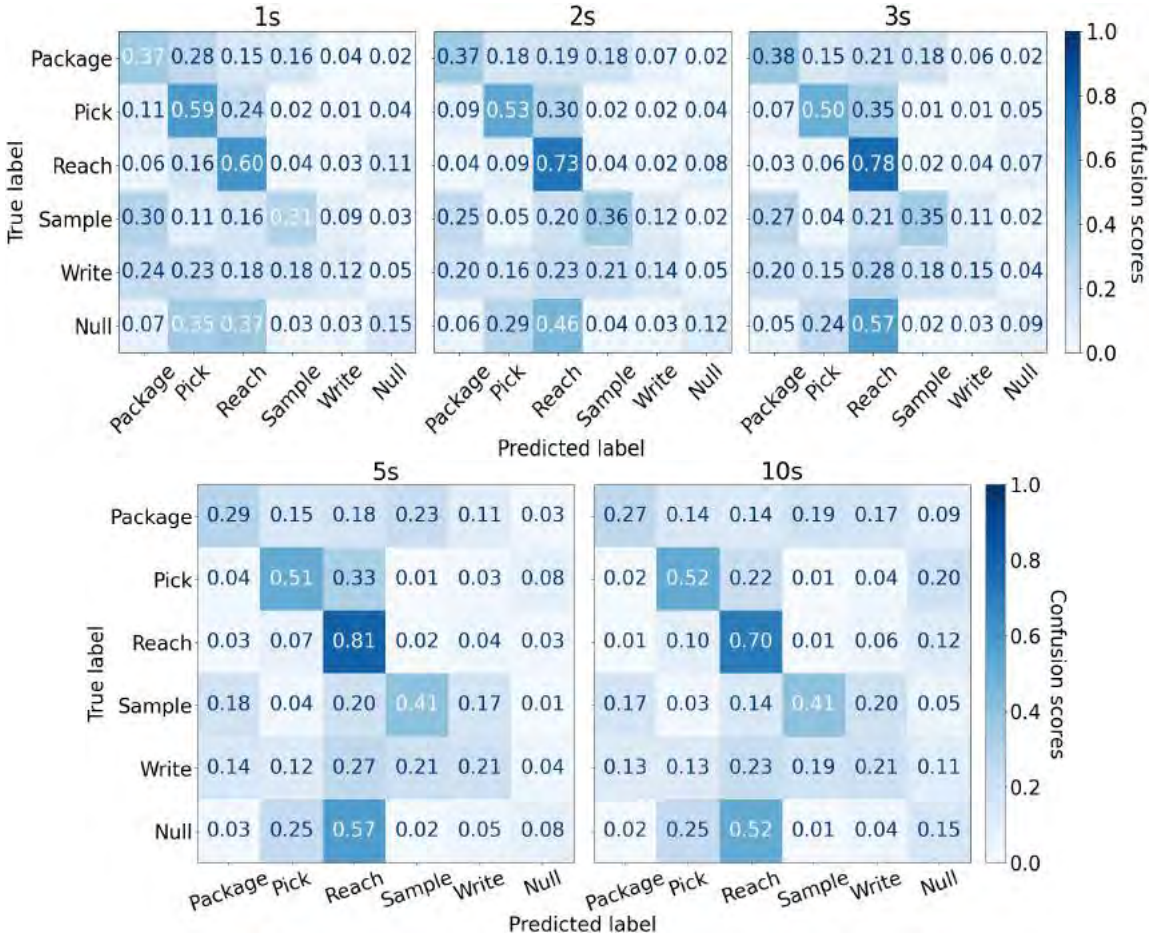


Figure 5.23: Fine-grained motor task recognition confusion matrices for the incorporated window sizes.

task recognition algorithms may be susceptible to task transitions due to signal variations [35]. The high confusion rates during transitions led to lower classification accuracy, as tasks switched frequently.

Overall, the 3s and 5s window sizes performed the best across both evaluations. The 3s is preferred over the 5s window size for detecting fine-grained motor tasks for the intended domain, due to its shorter duration. However, the tasks will have different durations and are highly sensitive to changes in window size. A short task (e.g., Package) may require a smaller window ($\leq 3s$) so that it is not confused with other tasks, while a long-duration task (e.g., Sample) may require a larger window ($\geq 5s$) to have sufficient context; therefore, it may be necessary to use an adaptive sliding window method [170, 195] to expand and contract the window size, based on the task. An ensemble learning algorithm may also be leveraged, where the algorithm makes predictions over multiple fixed window sizes and fuses the predictions across the window sizes intelligently to detect the tasks.

5.2.7 Tactile Task Recognition

The tactile task recognition algorithm incorporated inertial metrics provided by the Xsens sensors on the hands and the forearm 8-channel sEMG from the Myos to train a deep learning algorithm. The algorithm was trained to predict one of the eight tactile tasks: i) Grasp, ii) Hold item, iii) Lift, iv) Package, v) Press, vi) Sample, vii) Write, and viii) Null. The tactile tasks' data distribution before and after the downsampling process is presented in Table 5.16. Window sizes, $t_w = \{0.5s, 1s, 1.5s, 2s, 3s\}$, with a 50% overlap, were investigated for analyzing the window size's impact on the algorithm's performance. The shorter window sizes were chosen in order to be consistent with the prior supervisory evaluation (see Chapter 4.2.7).

The tactile algorithm's accuracy increased from 28.28% for the 0.5s window size to 31.76% for the 1s, but remained relatively the same beyond the 1s window size (see Figure 5.24). The Friedman's test identified a significant difference in accuracies between the window sizes ($\chi^2(4, 22) = 29.16, p < 0.01$). The Wilcoxon signed-rank test found that the 0.5s window size's accuracy was significantly lower than all other window sizes with a medium to large effect size ($p < 0.01$, $0.56 <$ Cohen's $d < 0.94$). No other differences were significant.

The 1s, 1.5s, 2s, and 5s window sizes' confusion matrices were analyzed to identify the best-performing window size (see Figure 5.25). The 0.5s window size's confusion matrix is

Table 5.16: The mean (std. dev.) and the cumulative task instances for the tactile component before and after downsampling, aggregated across participants.

Tasks	**Before Downsampling**		**After Downsampling**	
	Mean (std. dev.)	**Cumulative**	**Mean (std. dev.)**	**Cumulative**
Grasp	8483.00 (3056.92)	288422	3751.06 (1163.45)	127536
Hold item	4222.85 (1538.16)	143577	3645.03 (1266.81)	123931
Lift	2034.26 (827.14)	69165	2034.26 (827.14)	69165
Package	1918.82 (892.17)	65240	1917.32 (888.15)	65189
Press	5104.47 (2287.71)	173552	3369.35 (1146.07)	114558
Sample	3014.62 (1319.11)	102497	2930.21 (1223.50)	99627
Write	1495.59 (591.78)	50850	1495.59 (591.78)	50850
Null	8114.94 (2689.90)	275908	3751.06 (1163.45)	127536

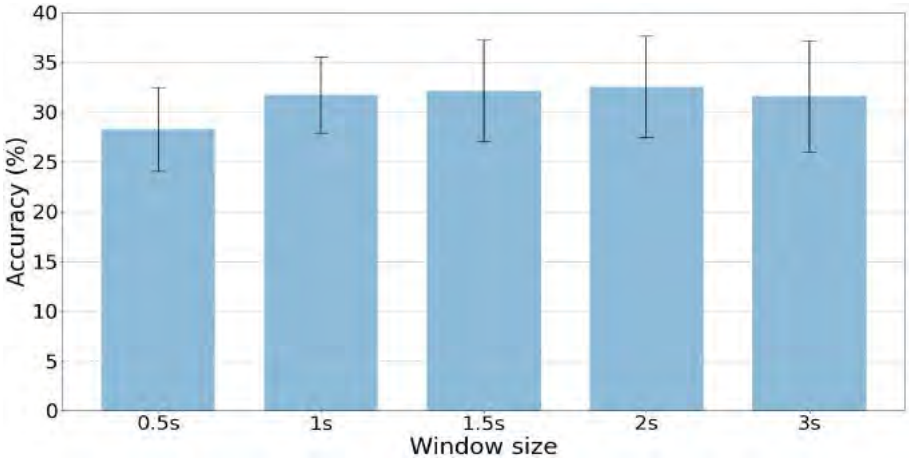


Figure 5.24: Tactile motor task recognition accuracy % mean (std. dev.) by window size with the Xsens' hand IMU and Myos' SEMG metrics from both arms.

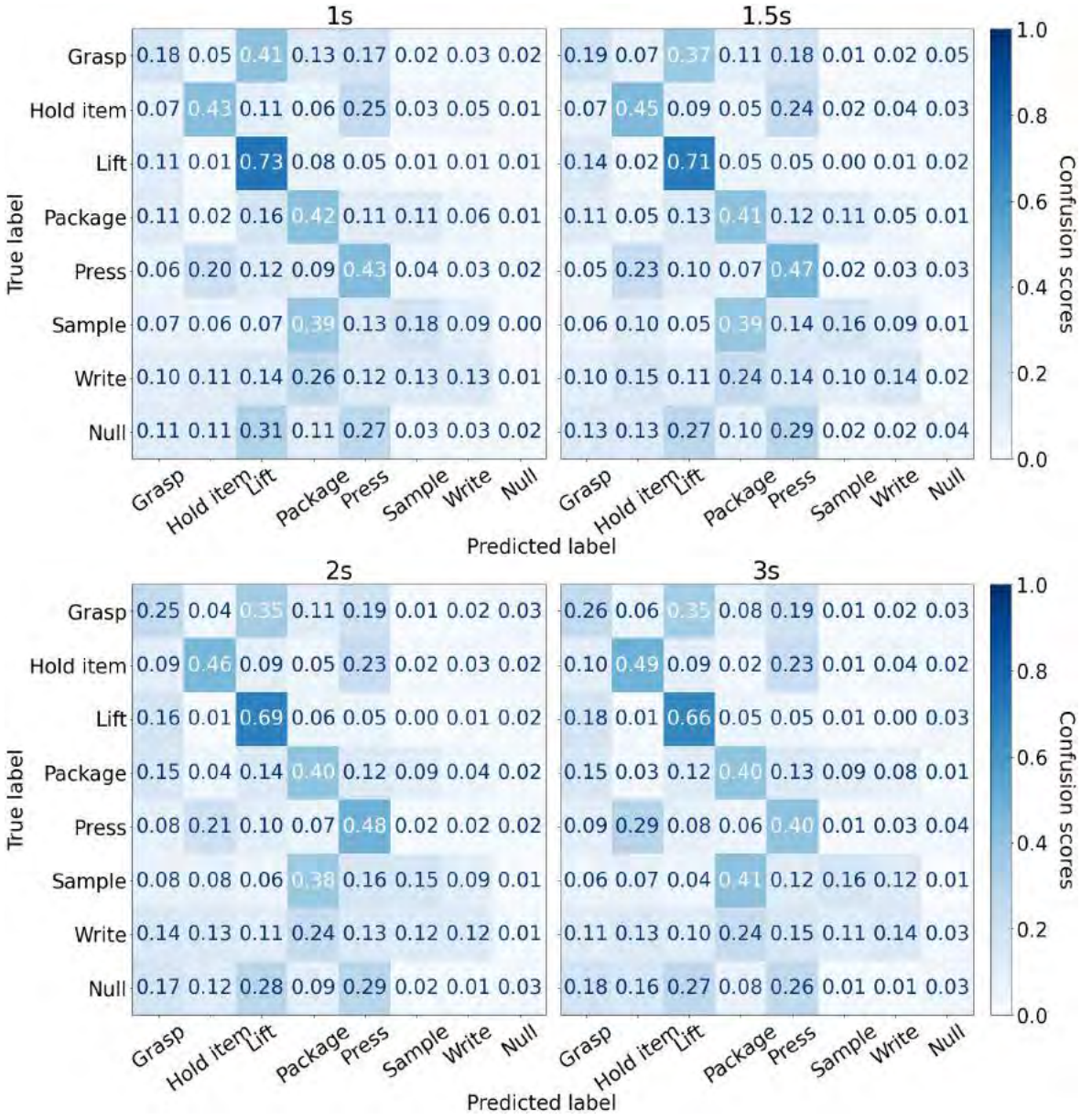


Figure 5.25: Tactile task recognition confusion matrices for the 1s, 1.5s, 2s, and 3s window sizes.

provided in Appendix B Figure B.4, as its overall accuracy was significantly lower than the rest. The algorithm had the highest classification rate for the Lift task ($\geq 66\%$), and the lowest for the Null task ($\leq 5\%$), followed by the Write task ($\leq 14\%$). All four window sizes had lower misclassification rates for the Hold item, Lift, Package, and Press tasks, while the other tasks had higher confusion rates. The Hold item task was often confused with the Press task and vice-versa. The Sample and Write tasks were often misclassified as the Package task, while the Grasp task was confused with the Lift task. The Grasp and Hold item tasks' classification rates increased with window size, while the Lift task's classification rate decreased with window size. The Press task's rate increased until the 2s window size, but dropped at the 3s window size. Other tasks' classification rates were unaffected by the change in window size. These observations indicate that some tactile tasks are sensitive to changes in window size, while others tend to be unaffected. Overall, the 1.5s and 2s window sizes' had lower misclassification rates for most tasks when compared to the 1s and 3s window sizes' misclassification rates.

5.2.7.1 Discussion

Hypothesis $\mathbf{H_1^T}$ predicted that the algorithm will detect tasks with $\geq 80\%$ classification accuracy for at least one of the window sizes, which was not supported. The algorithm's subpar performance for peer-based evaluation was primarily due to the low-data regime, caused by the increase in the number of tasks detected; however, the inter-task similarities (e.g., Hold item vs. Press) and task transitions (e.g., Sample vs. Write vs. Package) also had a significant impact.

The results from both evaluations suggested that detecting tactile tasks with high accuracy was difficult. The 1s window size was the recommended window size for the supervisory domain, with the 1.5s being a close second. The peer-based evaluation indicates that both 1.5s and 2s window sizes performed well. Thus, the 1.5s window size is the ideal trade-off for detecting tactile tasks across domains. Shorter window sizes were a logical choice for the supervisory domain, as most of the supervisory tactile interactions were momentary; however, the peer-based evaluation revealed that an adaptive sliding window or an ensemble learning approach is imperative to detect tactile tasks reliably, since tasks are sensitive to changes in window size.

5.2.8 GNN Fusion Task Consolidation

The *Fusion* algorithm refined the task recognition components' logged task detections by passing each component's individual algorithm's most recent task prediction scores as input to a GNN network to derive the logged task predictions across components via joint optimization (see Chapter 3.6). It is important to emphasize that the fusion algorithm optimized the components' logged task constituents, rather than the atomic tasks. The logged task constituents serve as a representation of the atomic tasks, demonstrating the fusion algorithm's ability to optimize the task detections across components in a highly dynamic, uncertain peer-based HRT task environment. This approach was necessitated by the availability of reliable ground truth data only for the logged task constituents. Nevertheless, the GNN fusion algorithm can jointly optimize the actual atomic tasks across components without sacrificing generality, provided the atomic tasks' reliable ground truth data is available (see Chapter 5.1.7).

The logged tasks identified for each activity component are summarized in Table 5.17. The gross motor component had two four tasks, the fine-grained motor had six, and the tactile had eight tasks. The visual and auditory components had five and ten tasks logged tasks, respectively, while the cognitive and speech components had seven tasks each. Thus, the GNN fusion algorithm consolidated forty-seven logged task detections and predicted the tasks based on the seven activity components (i.e., one per component) at any given instance.

Table 5.17: Logged tasks identified for each activity component across all mission and secondary tasks.

Component	**Logged tasks**
Gross motor	Walk, Bend over, Carry large object, Null
Fine-grained motor	Package, Pick, Reach, Sample, Write, Null
Tactile	Grasp, Hold, Lift, Package, Press, Sample, Write, Null
Visual	Coordination, Inspect, Locate, Scan, Null
Cognitive	Association, Conversation, Count, Evaluation, Process, Recall, Null
Auditory	Robot's analyze prompt, Robot's assist request, Robot's sample description request, Robot's sampling instructions, Robot's request to report to Incident Command, Incident Command communication, Incident Command reminder, In-situ probe, Secondary prompt, Null
Speech	Sample description, Suspicious item information, Incident Command information, Scan request, Secondary response, In-situ response, Null

5.2.8.1 Experimental Design

The consolidated logged detections can be fully correct, partially correct, or fully incorrect for a given instance. Therefore, accuracy per instance is calculated as the proportion of the predicted labels that are correct to the total number (i.e., predicted and true) of labels for a given instance. The *parital accuracy*, which is the average across all instances aggregated across participants (Equation 4.1), is used as the dependent variable to assess the fusion algorithm's overall performance.

Table 5.18: The individual algorithms and the corresponding window size and associated accuracy (mean % (std. dev.)) by component from the prior section that were employed by the fusion algorithm for consolidating the peer-based logged predictions.

Component	Algorithm	Window size	Accuracy
Cognitive	RF	15s	21.95 (4.31)
Speech	Deep learning	3s	47.19 (4.91)
Auditory	Deep learning	5s	47.95 (4.56)
Visual	RF	15s	45.65 (4.52)
Gross motor	Deep learning	3s	58.78 (8.59)
Fine-grained motor	Deep learning	3s	37.68 (4.46)
Tactile	Deep learning	1.5s	32.13 (5.12)

The peer-based evaluation's results indicate that the optimal window size differed for each individual algorithm across components (see Chapters 5.2.5 - 5.2.2). The tactile task recognition algorithm demonstrated its best performance with a window size of 1.5 seconds, while the visual task recognition algorithm excelled at $15s$ window size. The most effective window sizes for the remaining components, along with their respective accuracies, are outlined in Table 5.18. The GNN fusion algorithm utilized task predictions from each individual algorithm based on their respective best-performing window sizes to jointly optimize the logged task detections across components.

The GNN fusion algorithm was assessed using various window sizes $t_w = 1s$, $3s$, $5s$, $10s$, $15s$, employing a one-and-half-second stride (i.e., $t_s = 1.5$s) to inform the window size's impact on the GNN fusion algorithm's performance. The evaluated window sizes encompass the range of window sizes used in the best-performing individual task component algorithms. The maximum window size, $15s$, was utilized for the visual and cognitive components. The stride duration was determined by selecting the shortest duration required to make a logged task prediction, which amounted to 1.5 seconds for the tactile component.

5.2.8.2 Results

Overall, the fusion algorithm's partial accuracy for the 1s window size was 87.01% and dropped to 83.02% at the 3s window size, and remained relatively the same until the 15s window size (see Figure 5.26). The Friedman's test indicated a significant difference between window sizes ($\chi^2(4, 36) = 13.40, p < 0.01$). The Wilcoxon signed-rank test found that the 1s window size's partial accuracy was significantly higher than all other window sizes with a medium effect size ($p < 0.01$, $0.56 <$ Cohen's $d < 0.89$). No other differences were significant. A detailed examination revealed that the 1s variant exhibited a notable bias towards predicting the *Null* task for the visual and cognitive components. This bias inflated the 1s variant's overall accuracy, primarily due to the high frequency of instances associated with this task. Consequently, when considering the bias associated with the 1s window size, the fusion algorithm's partial accuracies did not show significant variations across window sizes.

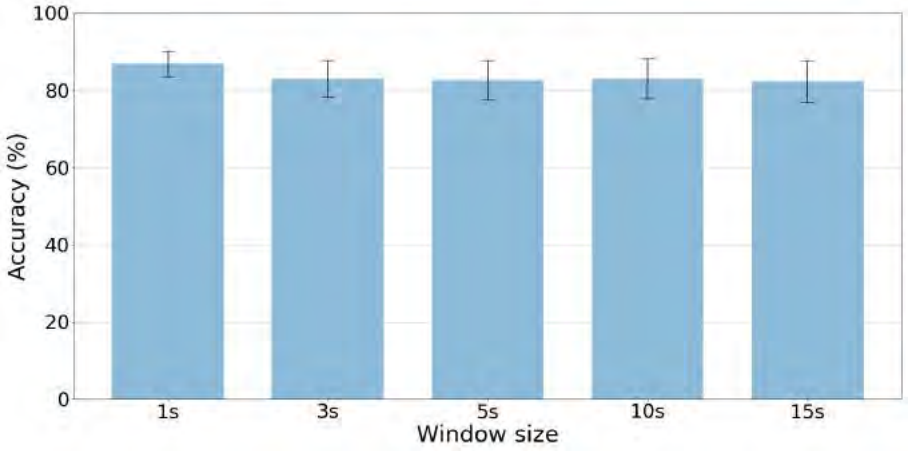


Figure 5.26: GNN fusion algorithm's partial accuracy % by window size aggregated across participants.

The fusion algorithm predicted seven logged tasks simultaneously, each associated with one of the seven activity components; therefore, each component's accuracy improvements achieved by the fusion algorithm can be compared against its corresponding best-performing individual algorithm, as illustrated in Figure 5.27. Overall, the GNN fusion algorithm can detect the logged tasks with $\geq 60\%$ accuracy across all components. The components' accuracies improved by $30 - 40\%$, with the visual component experiencing

the most substantial improvement (40%) and tactile being the least improved component (30%). The GNN fusion's joint optimization also led to an increase in accuracy variability for activity components with a larger number of tasks (i.e., fine-grained motor, tactile, auditory, and speech) by $10 - 18\%$. The Wilcoxon signed-rank test indicated that the accuracies post GNN fusion's joint optimization were significantly higher than the corresponding individual algorithm's accuracies across all components ($p < 0.01$). Figure 5.27 indicates that the fine-grained motor, tactile, and cognitive components bottlenecked the GNN fusion algorithm's performance. The GNN fusion algorithm's 5s window size had the best overall performance, as it achieved the highest improvement or the second highest across most components.

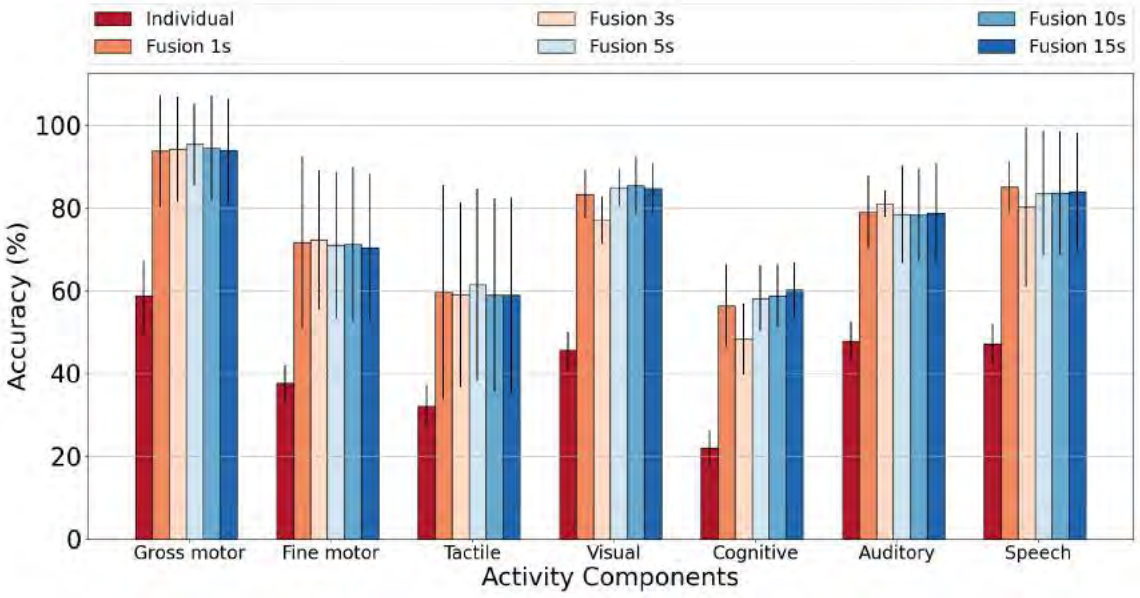


Figure 5.27: The accuracy (mean % (std. dev.)) comparisons between the individual algorithms and the GNN fusion algorithm by activity components for the evaluated window sizes. NOTE: Each component's individual algorithm's accuracy corresponds to its best-performing window size's accuracy.

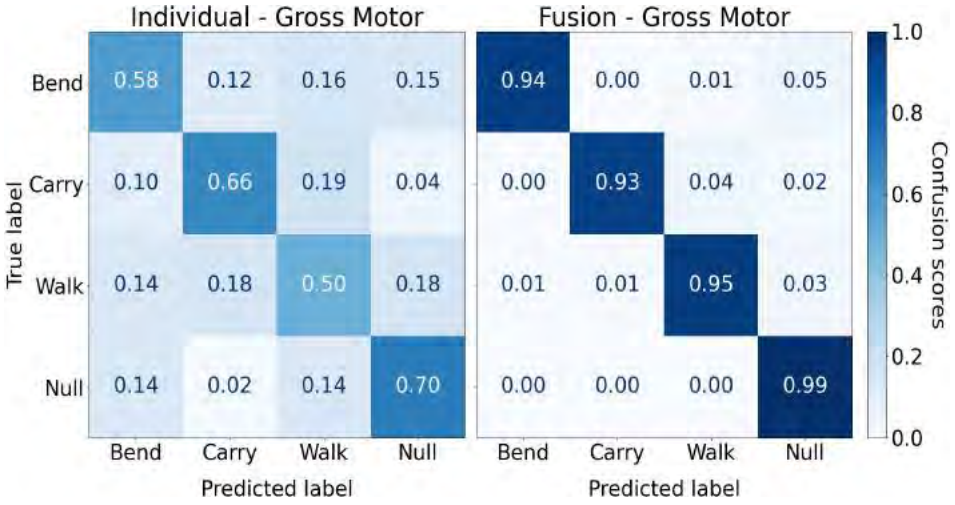


Figure 5.28: Gross motor component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (5s window size).

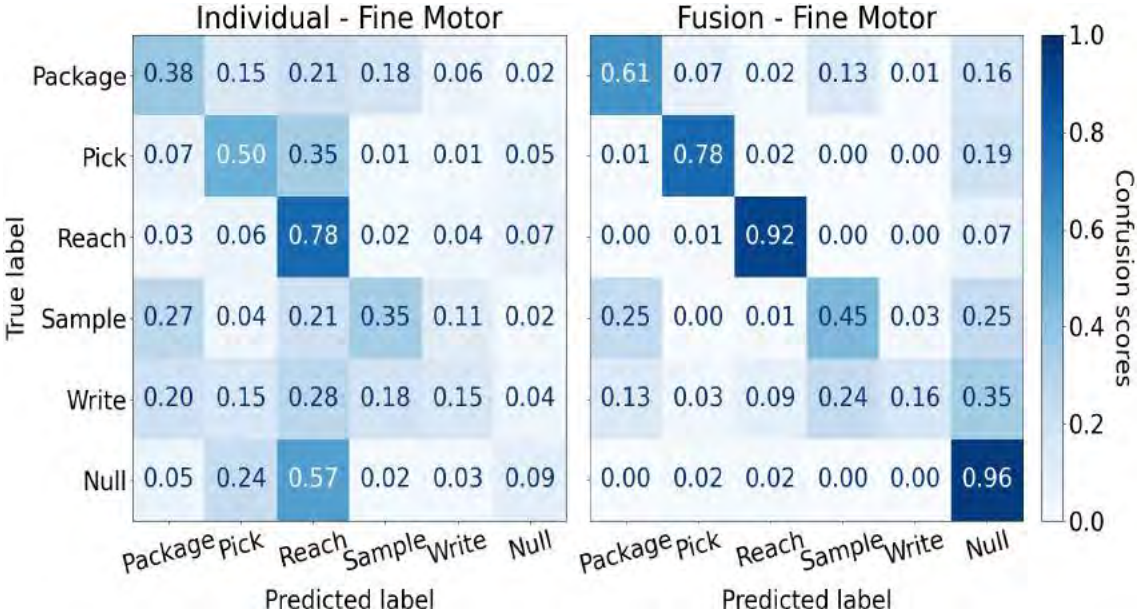


Figure 5.29: Fine motor component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (5s window size).

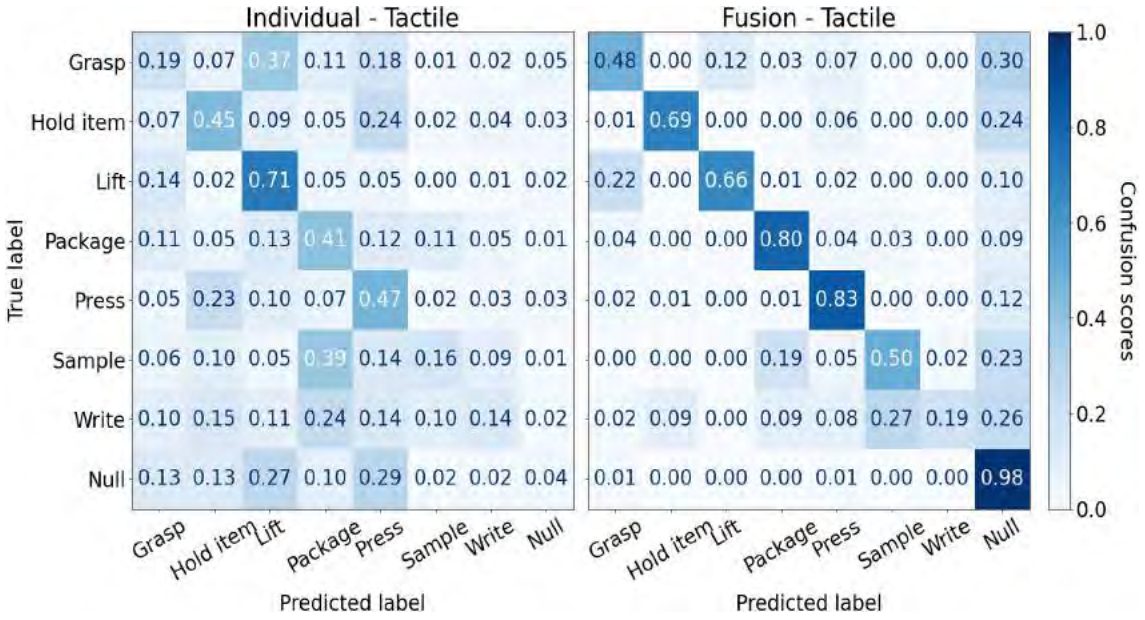


Figure 5.30: Tactile component's confusion matrix for its best-performing individual algorithm (1.5s window size) vs. GNN fusion algorithm (5s window size).

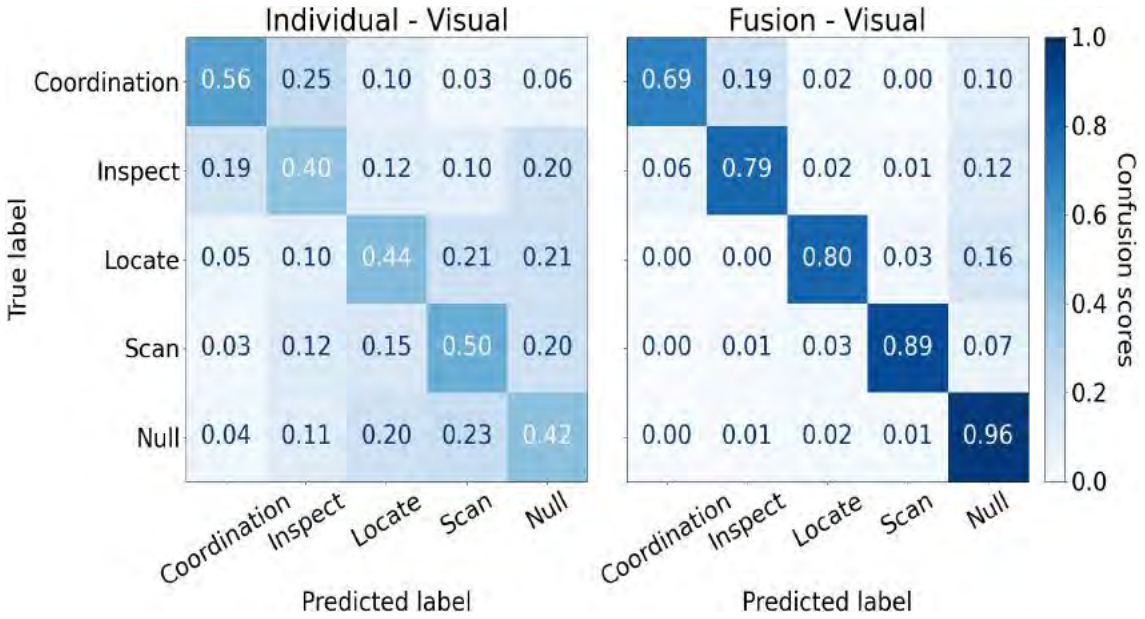


Figure 5.31: Visual component's confusion matrix for its best-performing individual algorithm (15s window size) vs. GNN fusion algorithm (5s window size).

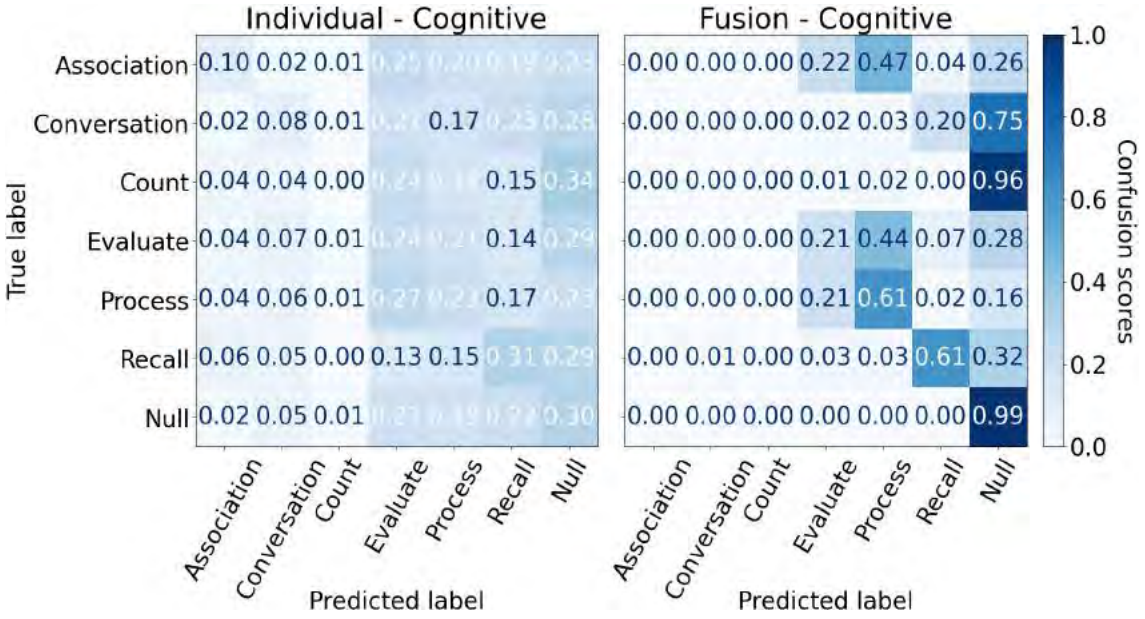


Figure 5.32: Cognitive component's confusion matrix for its best-performing individual algorithm (15s window size) vs. GNN fusion algorithm (5s window size).

The confusion matrix of each component's best-performing individual algorithm is compared against the corresponding confusion matrix derived from the 5s window size GNN fusion algorithm. This comparison analyzes the task accuracies across components pre (indicated as individual in the respective figures) and post (indicated as fusion in the respective figures) GNN fusion's joint optimization (see Figures 5.28, 5.29, 5.30, 5.31, 5.32, 5.33, 5.34). The confusion matrices indicate that the *Null* task was detected with near perfection (i.e., $\geq 91\%$ accuracy) across all components. The GNN fusion's joint optimization increased the recognition rate for most tasks across components by $7-36\%$, with the gross motor, visual, and speech components attaining the highest recognition rate improvement by tasks. The GNN fusion algorithm also led to a decrease in the recognition rate for a majority of the cognitive tasks, along with a few tactile and auditory tasks (e.g., the tactile's lift task and the auditory's Incident Commander secondary prompt task).

Both the fine-grained motor and tactile components' Sample and Write tasks continued to be misclassified as Package and Sample, respectively, even after the GNN fusion's joint optimization (see Figures 5.29 and 5.30). These instances of confusion highlight that inertial-based task recognition algorithms are susceptible to task transitions. While the

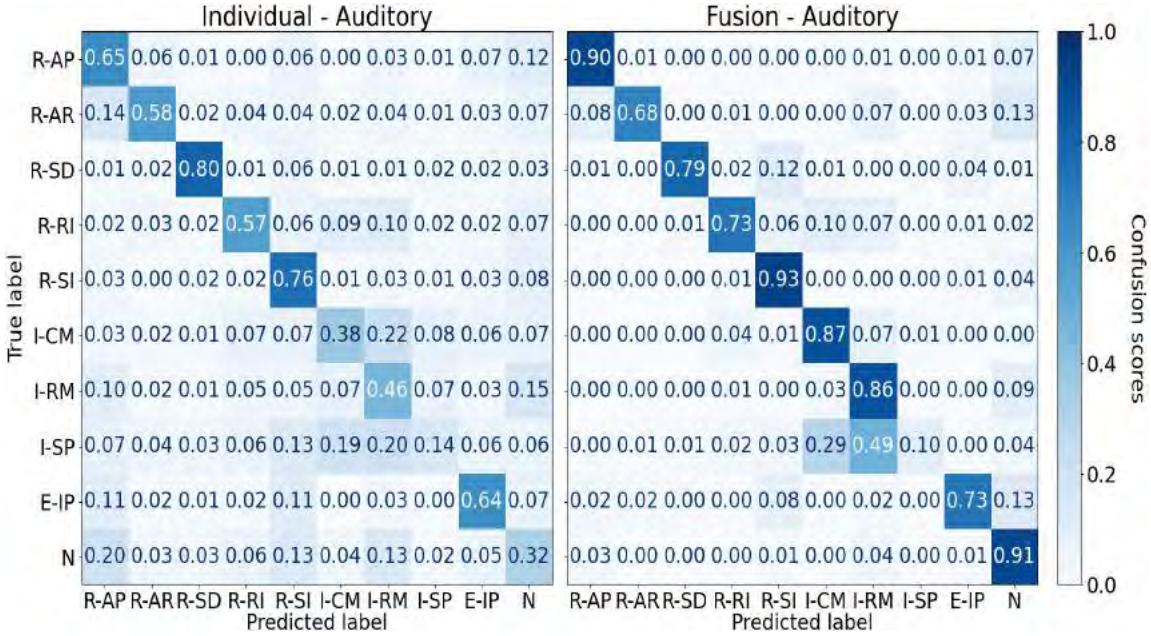


Figure 5.33: Auditory component's confusion matrix for its best-performing individual algorithm (5s window size) vs. GNN fusion algorithm (5s window size). Reminder: *R-AP*: Robot's analyze prompt, *R-AR*: Robot's assist request, *R-SD*: Robot's sample description request, *R-RI*: Robot's report to Incident Commander prompt, *R-SI*: Robot's sampling instructions, *I-CM*: Incident Commander's communication, *I-RM*: Incident Commander's reminder, *I-SP*: Incident Commander's secondary prompt, *E-IP*: Experimenter's in-situ probe, and *N*: Null.

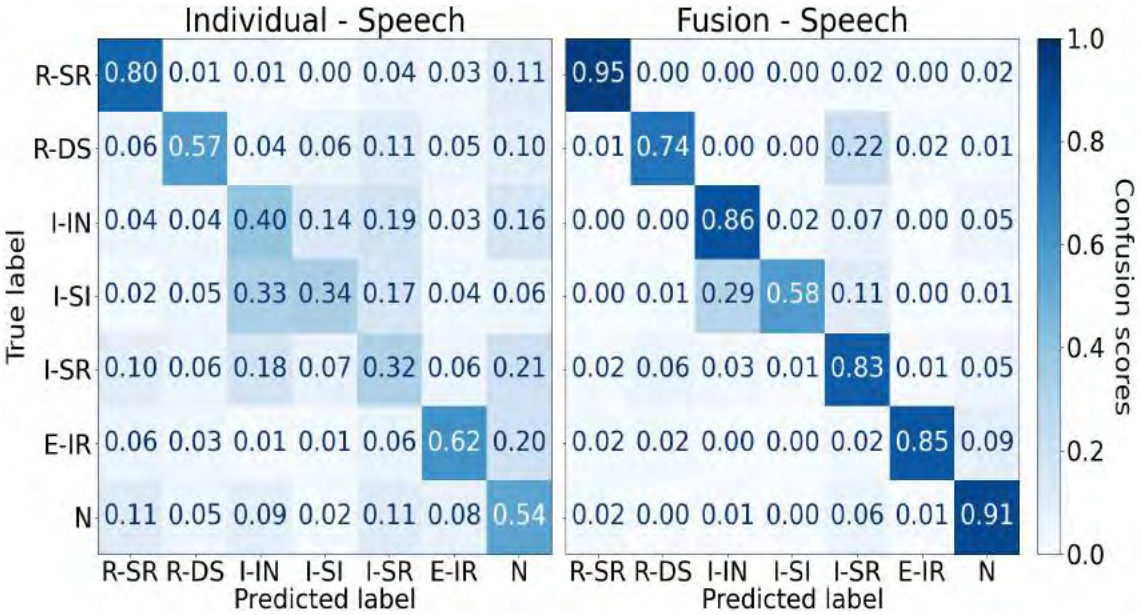


Figure 5.34: Speech component's confusion matrix for its best-performing individual algorithm (3s window size) vs. GNN fusion algorithm (5s window size). Reminder: *R-SR*: Requesting robot to scan an item, *R-DS*: Describing sample to the robot, *I-IN*: Providing information to the Incident Commander, *I-SI*: Describing a suspicious item to the Incident Commander, *I-SR*: Responding to Incident Commander's secondary prompt, *E-IR*: Responding to experimenter's in-situ probe, and *N*: Null.

fusion algorithm improves the tasks' recognition rate, there are limits to its effectiveness in preventing such occurrences. A similar trend was observed for the auditory component, where the Incident Commander's secondary prompts were still misclassified as reminders and communications, even after joint optimization, likely due to the high similarity between these conversational events (see Figure 5.33). The cognitive component's individual algorithm exhibited a strong bias toward predicting the Evaluate, Process, Recall, and Null tasks in its pre-fusion predictions, which led to elevated misclassification rates for the Association, Conversation, and Count tasks (as shown in Figure 5.32). The fusion algorithm's joint optimization improved the recognition rates of the Process and Recall tasks, but compromised the accuracy for other cognitive tasks due to the bias toward the Null task. Interested readers can refer to Appendix B Chapter B.5 for the rest of the GNN fusion algorithm's window sizes' confusion matrices.

Overall, most of the gross motor, visual, speech, and auditory components' tasks were detected with $\geq 80\%$ or $\sim 80\%$ accuracy post GNN fusion's joint optimization. The fine-grained motor and tactile components' tasks achieved intermediate accuracy, ranging from $50 - 85\%$ for most tasks, while the cognitive tasks were not detected reliably.

5.2.8.3 Discussion

The GNN fusion algorithm demonstrated high sensitivity across all window sizes; however, it is important to note that the partial accuracy may have been artificially boosted due to the long-tailed data distribution. Across all activity components, a long-tailed distribution was observed, where specific tasks, particularly the *Null* task, constituted over 80% of the data, while other tasks were under-represented. This dominance of *Null* tasks influenced the fusion algorithm when aggregated across all seven components, potentially leading to an inflated partial accuracy.

Hypothesis $\mathbf{H_2}$ predicted that the GNN fusion algorithm's joint task optimization will improve the atomic task detection accuracy to $\geq 80\%$ across all seven components, which was only partially supported. The fusion algorithm's effectiveness hinges on the accuracy of the individual algorithms' predictions. Although the fusion algorithm can reduce a majority of misclassifications across components, those that remained cannot be avoided entirely and were perpetuated by task transitions (e.g., fine-grained motor and tactile), and inter-task similarities (e.g., auditory). Furthermore, the cognitive task detections made by the fusion algorithm were unreliable, due to the subpar predictions generated by

the cognitive component's individual task recognition algorithm.

The GNN fusion algorithm is a viable option for consolidating atomic task predictions across the supervisory (see Chapter 4.2.8) and peer-based domains. Determining an ideal window size for the fusion algorithm remains challenging, as different window sizes performed better for different components. The 5s window size is recommended as it demonstrated the most favorable overall performance across components. Future work must investigate incorporating adaptive or ensemble window size prediction methods for the fusion algorithm.

5.2.9 Composite and Concurrent Task Recognition

The TCN-based *Composite and Concurrent* task recognition algorithm (as detailed in Chapter 3.7), utilized the atomic task time series $\mathbf{X}$ as input to predict a set of twenty-one composite tasks, each involving multiple activity components. The breakdown of these tasks can be found in Chapter 5.1.7. Eighteen of the twenty-three composite tasks stem from the six mission tasks, while the remaining three are associated with secondary tasks (see Table 5.8).

The TCN-based algorithm identified concurrent composite tasks (i.e., ≥ 1 composite tasks at any given instance) by estimating the probability of each composite task's existence for a given atomic task time series. The algorithm's composite task predictions are classified into three categories: fully correct, partially correct, or fully incorrect. Two dependent variables, *exact match ratio* (Equation 4.2) and *partial accuracy* (Equation 4.1), along with multi-label confusion matrices were employed to assess the performance of the TCN-based algorithm.

The input time series $\mathbf{X}$'s temporal duration was varied using different window sizes $t_w = 1s$, $3s$, $5s$, $10s$, $15s$, with a one-and-half second stride (i.e., $t_s = 1.5s$) to inform its impact on the TCN algorithm's performance. The evaluated window sizes span the range examined for the fusion algorithm, while the stride duration was set to match the shortest window size used across all components.

Overall, the algorithm's exact match ratio for the 1s window size was 80.30% and gradually increased until the 15s window size (86.89%) (see Figure 5.35). The exact match ratio's std. dev. was relatively low ($< 3\%$) across window sizes, indicating that the algorithm's accuracy increases with window size, while maintaining the precision level. The Friedman's test indicated a significant difference between window sizes ($\chi^2(4, 36) = 122.42, p < 0.01$).

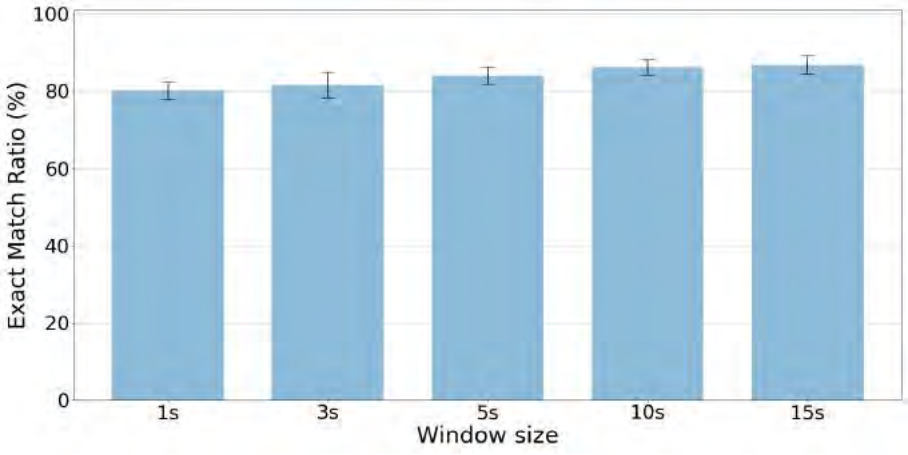


Figure 5.35: TCN composite and concurrent task recognition algorithm's exact match ratio % mean (std. dev.) by window size aggregated across participants.

The Wilcoxon signed-rank test found that the 15s window size's exact match ratio was significantly higher than all other window sizes ($p < 0.01$, $0.30 <$ Cohen's $d < 2.82$), while the 1s window size's was significantly lower than the others with a large effect size ($p < 0.01$, $0.48 <$ Cohen's $d < 2.82$). The 10s window size's exact match ratio was significantly higher than the 3s and 5s ($p < 0.01$, $0.99 <$ Cohen's $d < 1.67$), while the 5s window size's exact match ratio was significantly higher than the 3s ($p < 0.01$, Cohen's $d = 0.86$).

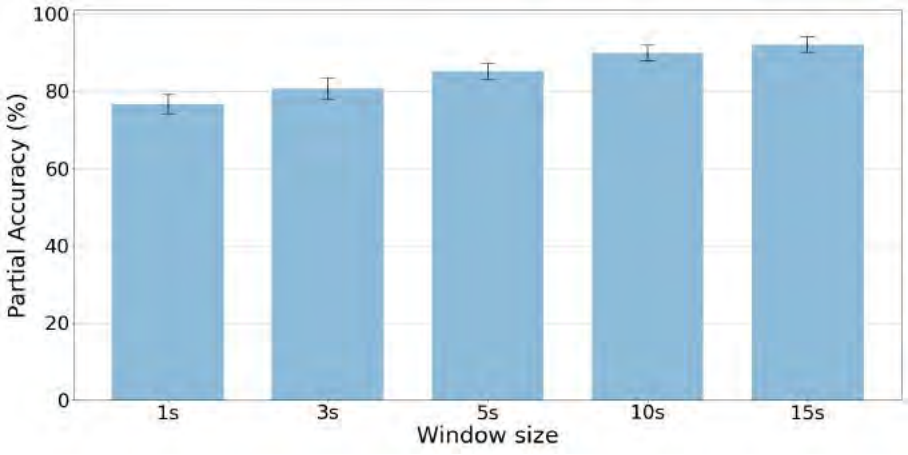
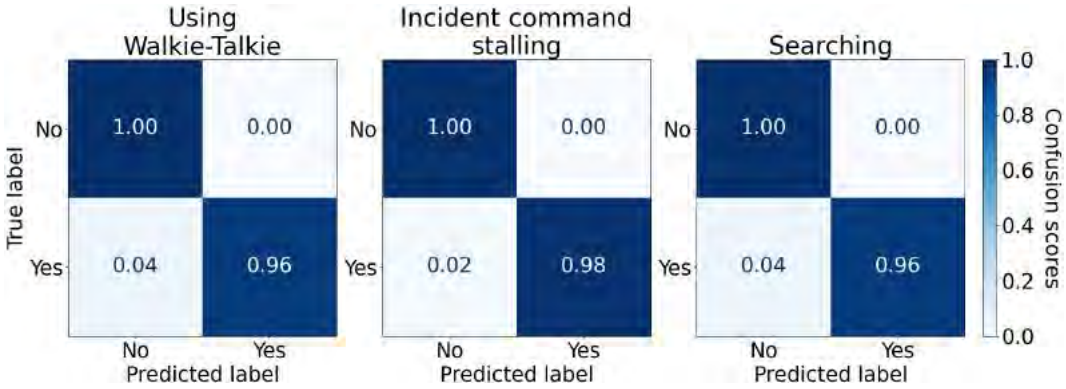
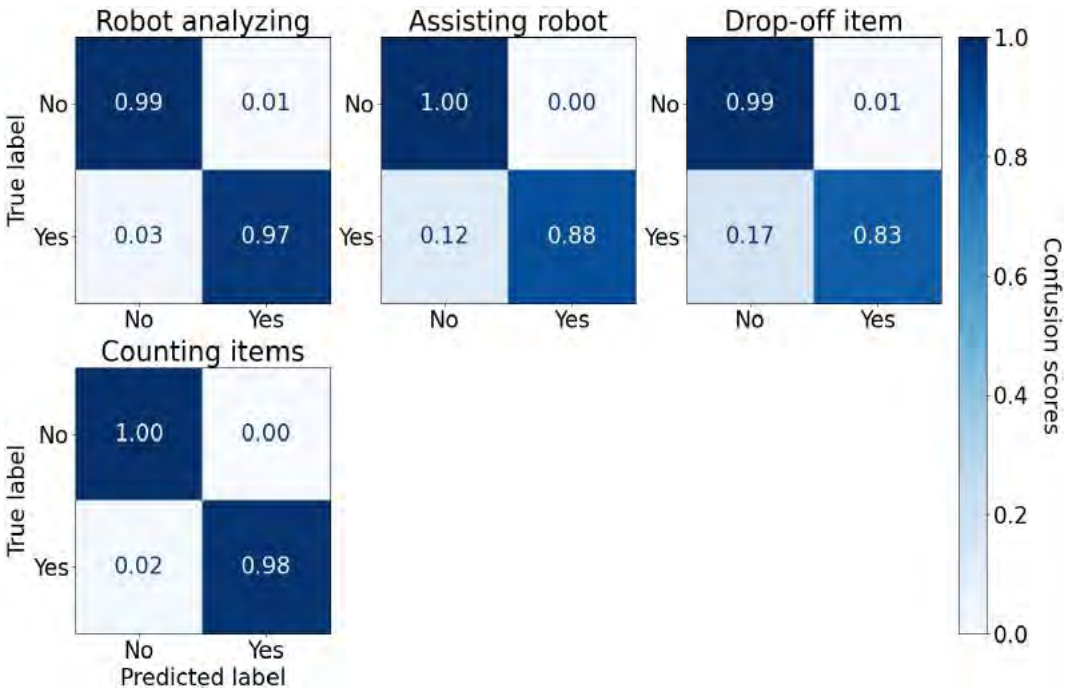


Figure 5.36: TCN composite and concurrent task recognition algorithm's partial accuracy % mean (std. dev.) by window size aggregated across participants.

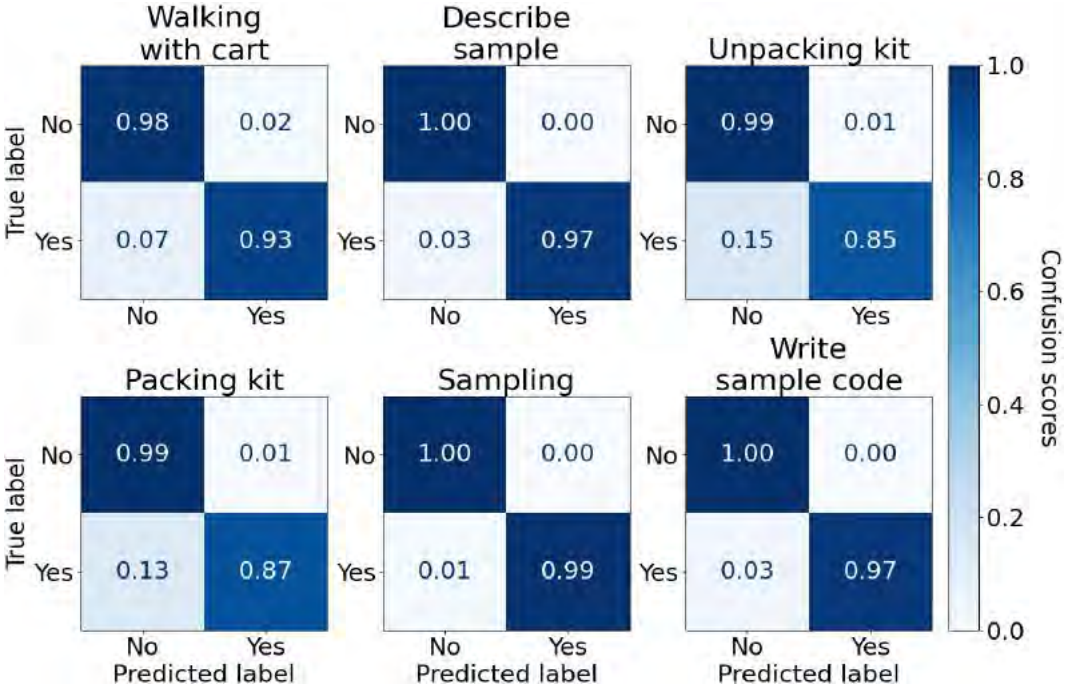
The algorithm's partial accuracy for the 1s window size was 76.75%, which gradually increased until the 15s window size (92.20%) (see Figure 5.36). The partial accuracy's std. dev. was relatively low ($< 3\%$) across window sizes, following a similar trend to the algorithm's exact match ratio. The Friedman's test indicated a significant difference between window sizes ($\chi^2(4, 36) = 141.67, p < 0.01$). The Wilcoxon signed-rank test found that the 15s window size's partial accuracy was significantly higher than all other window sizes ($p < 0.01$, $1.07 <$ Cohen's $d < 6.80$), while the 1s window size's was significantly lower than the rest, with a very large effect size ($p < 0.01$, $1.55 <$ Cohen's $d < 6.80$). The 10s window size's partial accuracy was significantly higher than the 3s and 5s ($p < 0.01$, $2.25 <$ Cohen's $d < 3.81$), while the 5s window size's partial accuracy was significantly higher than the 3s ($p < 0.01$, Cohen's $d = 1.77$). The results indicated that the algorithm's performance in terms of both exact match ratio and partial accuracy increased with the temporal window size, with the 15s window size significantly outperforming the rest.



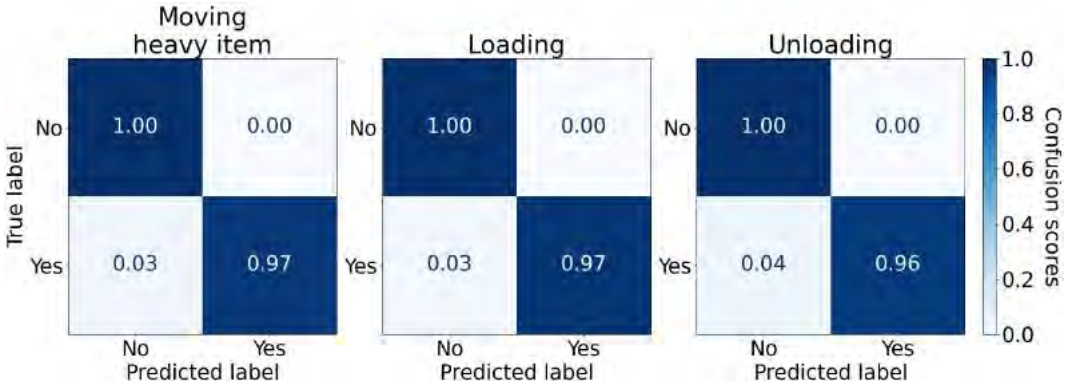
(a) Multi-label confusion matrices for the composite tasks that were shared across missions.



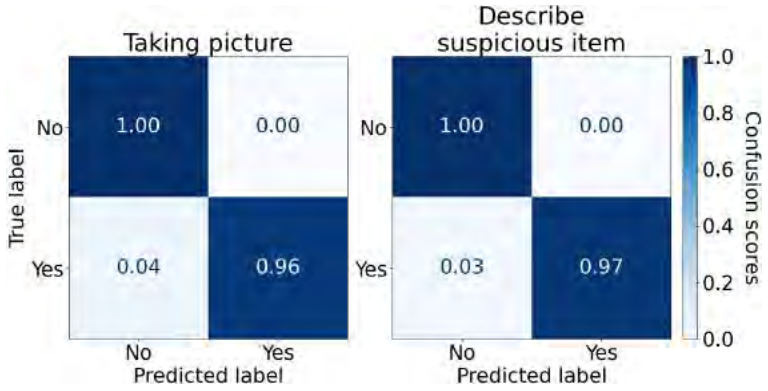
(b) Multi-label confusion matrices for the Pharmacy and Pawnshop missions' composite tasks.



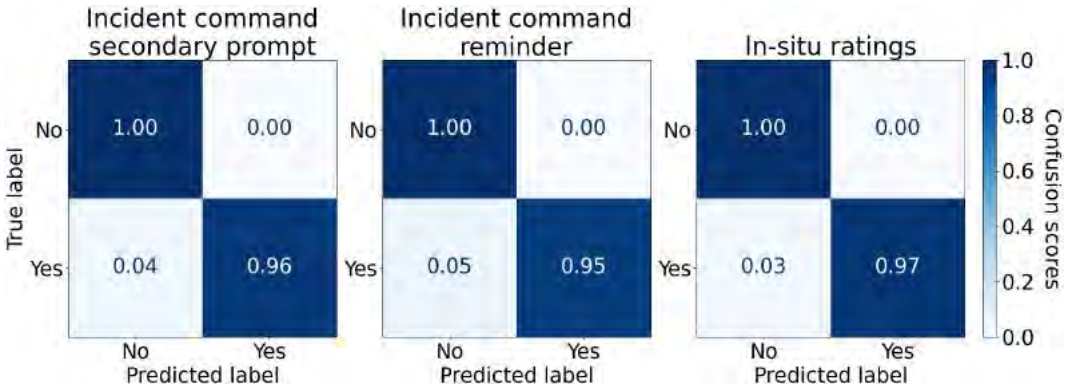
(c) Multi-label confusion matrices for the Solid and Liquid sampling missions' composite tasks.



(d) Multi-label confusion matrices for the Debris mission's composite tasks.



(e) Multi-label confusion matrices for the Search mission's composite tasks.



(f) Multi-label confusion matrices for the Secondary composite tasks.

Figure 5.37: The TCN algorithm's 15s window size variant's multi-label confusion matrices grouped by mission and secondary tasks aggregated across participants.

The composite tasks' multi-label confusion matrices were analyzed for the 15s window size by clustering the composite tasks by missions (see Figure 5.37). The TCN-based algorithm detected the composite tasks' absence (i.e., true negatives) ($> 98\%$) and presence (i.e., true positives) ($\geq 83\%$) with high sensitivity across all mission and secondary tasks. Among the composite tasks, the *Drop-off item* and *Assisting robot* had relatively lower true positive rates of 83% and 88%, respectively (as shown in Figure 5.37b), which can be attributed to the tasks' shorter duration as compared to the rest. Similarly, the *Packing* and *Unpacking kit* composite tasks also exhibited lower true positive rates (as seen in Figure 5.37c) due to their shared underlying atomic task patterns. Most of the other composite tasks had true positive rates of 95% or higher.

Other window sizes had intermediate performances and the confusion matrices are provided in Appendix B Chapter B.6.

5.2.9.1 Discussion

Hypothesis $\mathbf{H_3}$ states that the TCN task recognition algorithm will detect concurrent composite tasks with $\geq 80\%$ accuracy, which was supported for the $\geq 5s$ window sizes. The results demonstrate the algorithm's ability to detect composite tasks reliably for the peer-based domain. The algorithm's exact match ratio (%) and partial accuracy did not reach a saturation point and continued increasing, which indicates that the TCN's dilated causal convolutions benefit from the increased temporal context as the window size expands. The 15s window size is the recommended window size for detecting concurrent composite tasks for the peer-based domain, as it is relatively shorter in duration, contains a lower number of trainable parameters, and exceeds the expected performance.

Chapter 6: Conclusion

HRTs collaborating to achieve tasks under various conditions, especially in unstructured, dynamic environments will require robots to adapt autonomously to a human teammate's state. An essential element of such adaptation is the robot's ability to infer the human teammate's current tasks, since understanding human's actions and their effects on the world provides the robot with the necessary context for assisting the human. Existing task recognition algorithms can detect tasks involving at most four activity components. This dissertation developed a multi-dimensional task recognition algorithm to detect tasks across contributing components: cognitive, visual, speech, auditory, gross motor, fine-grained motor, and tactile. The developed algorithm fused the components' individual task predictions intelligently in order to recognize the concurrent, composite tasks. The algorithm's performance was validated using data collected from supervisory-based and peer-based human-machine teaming evaluations, demonstrating that the developed task recognition algorithm can be applied across task domains and teaming roles.

6.1 Cross HRT-Role Discussion

This dissertation examined three hypotheses to analyze the multi-dimensional task recognition algorithm's ability to detect atomic and composite tasks reliably across two HRT domains. The outcomes of these hypotheses are outlined in Table 6.1. Overall, the individual task recognition algorithms' accuracy and reliability in detecting each component's atomic (or logged) tasks were lower. Several factors contributed to this underperformance: task transitions, inter-task similarity, intra-task variability, and individual differences. The algorithm's performance for the peer-based evaluation was exacerbated by the increased number of tasks and low-data regime.

Identifying sensitive metrics is of paramount importance; however, even more critical is segmenting those metrics effectively using appropriate window sizes to ensure they encompass sufficient information for task detection. The analyses revealed that a universal window size for task detection does not exist, as the optimal window size varied between components. For example, algorithms that incorporated inertial metrics (e.g., gross motor,

Table 6.1: Overall hypotheses summary. NOTE: ✓, – , and ✗ indicate full support, partial support, and no support, respectively.

Hypothesis	**Component**	**Evaluation**	
		Supervisory	**Peer-based**
Individual algorithms will detect tasks with $\geq 80\%$ classification accuracy.	Gross motor	✓	✗
	Fine-grained motor	✗	✗
	Tactile	✗	✗
	Visual	✗	✗
	Cognitive	✗	✗
	Auditory	–	✗
	Speech	✓	✗
The GNN fusion algorithm's joint task optimization will improve the atomic task detection accuracy to $\geq 80\%$ across all seven components.	N/A	✓	–
The TCN task recognition algorithm will detect concurrent composite tasks with $\geq 80\%$ accuracy.	N/A	✓	✓

fine-grained motor, and tactile) demonstrated superior performance with smaller windows ($\leq 5s$), while those reliant on eye tracking metrics (e.g., visual) necessitated larger windows ($\geq 10s$). Additionally, within each component, the optimal window size fluctuated depending on the task's duration, and the dynamic characteristics of the task environment, as detailed in Chapter 5.2. Overall, determining the optimal window size for a component hinges on several factors (e.g., the incorporated metrics, the tasks being detected, and the intended task environment's dynamicity). An alternative approach is to avoid using a fixed sliding window size methodology and instead opt for adaptive or ensemble sliding window methods to improve the individual task recognition algorithms' performance.

Each component's individual algorithm provided task predictions independently; however, both supervisory and peer-based evaluations revealed that the components' task predictions are highly correlated. The GNN fusion algorithm leveraged this interdependency to improve the atomic task predictions across components. The fusion algorithm played a crucial role in compensating for the individual algorithms' subpar performance via joint optimization. However, it is important to note that the joint optimization's effectiveness is contingent on the individual algorithms' performance, as demonstrated in Chapter 5.2.8, where although each component's recognition rate increased significantly, three out of the

seven components' accuracy did not meet the $\geq 80\%$ threshold.

Nevertheless, the fusion algorithm serves as a crucial link between the independent individual task recognition algorithms and an adaptive HRT system by providing reliable, accurate atomic task predictions. For example, a workload estimation algorithm may utilize these atomic predictions to inform each component's workload estimates, or use this information to generate a more accurate overall workload estimate by assigning weights to each component's workload levels based on the atomic task predictions. This refined overall estimate can enable the adaptive HRT system to make more informed decisions about how an adaptation will affect the human teammate.

The TCN-based algorithm detected the concurrent, composite tasks with high sensitivity across both evaluated domains, provided sufficient temporal context was available. Composite tasks typically require a longer duration to complete, as they involve a sequence of coordinated atomic and sub-composite tasks performed across components. Detecting both composite and atomic tasks simultaneously can enable an adaptive system to understand the specific composite task the human is currently engaged in, as well as gauge their progress towards completing it. This insight can be extremely useful for projecting the human teammate's future workload levels. An adaptive system can leverage this information to optimize the team's task priorities and allocations, maximizing the team's overall performance and collaboration.

6.1.1 Multi-Dimensional Task Recognition Algorithm Evaluation

This dissertation established six criteria for evaluating task recognition algorithms: *sensitivity*, *suitability*, *generalizability*, *composite factor*, *concurrency*, and *anomaly awareness* (see Chapter 2.3). While each component's individual task recognition algorithm fell short of the expected 80% accuracy threshold, the GNN fusion's joint optimization resulted in $\geq 80\%$ accuracy for all seven components' supervisory tasks and four out of the seven components' peer-based tasks. Thus, the developed multi-dimensional task recognition algorithm demonstrated medium to high sensitivity in detecting tasks across components. The algorithm conformed to the suitability criterion by incorporating sensors that are not environmentally embedded for task detection. The algorithm only partially conformed to the generalizability criterion, as it did not achieve the 80% accuracy threshold for all peer-based tasks, despite being evaluated using the *leave-one-subject-out* cross-validation. Finally, the multi-dimensional task recognition algorithm conformed to the composite fac-

tor and concurrency criteria, but did not meet the anomaly awareness criterion, as it did not detect out-of-class instances.

The developed multi-dimensional composite task recognition architecture is viable for detecting atomic and composite tasks across components. These algorithms can be integrated into an adaptive HRT system, enabling robots to adapt to the state of their human teammates; however, it is important to acknowledge the limitations. Chapter 6.2 summarizes the dissertation's contribution to the field, while Chapter 6.3 outlines the drawbacks and addresses measures to overcome them as future research directions.

6.2 Contributions

This dissertation resulted in three contributions to the field, each of which are summarized below:

1. This dissertation is the first to recognize tasks across the seven activity components using wearable sensors viable for unstructured dynamic domains. Two algorithms [75, 97] come the closest by identifying tasks across four activity components, but none detected tasks belonging to all seven activity components using wearable sensors. The multi-dimensional task recognition algorithm detected tasks across human-robot teaming paradigms by identifying and incorporating metrics that are sensitive, versatile, and suitable to employ across unstructured task environments. Multiple task environments were used to validate the algorithm's ability to detect tasks in representative domains. These environments comprised a diverse set of tasks with varying complexity levels, demonstrating that the algorithm was not constrained to a specific task environment.

2. The developed fusion algorithm combines the task detections across the seven components using GNNs to infer the atomic tasks indirectly for components with subpar task recognition accuracy. Robots need a holistic understanding of tasks' specific individual task recognition components to detect them accurately, since different tasks require different combinations of the activity components. The fusion algorithm can enable a system to leverage its understanding of an individual's complete task engagement state across components to intelligently target adaptations based on this knowledge (note, adaptation was outside the scope of this dissertation).

3. This dissertation is the first to develop a task recognition algorithm that utilizes wearable sensors for identifying composite tasks involving multiple activity components. A small number of composite task recognition algorithms exist, but are typically limited to detecting composite tasks containing only gross and fine-grained motor components (e.g., [161, 165, 185]). HRTs often perform a wide variety of tasks involving combinations of all activity components.

   Further, the developed composite task recognition can detect concurrent composite tasks. Existing task recognition literature typically assumes that an individual only performs one task at a time, which is not the case for many HRT scenarios, where the human may perform two or more tasks concurrently. A robot's ability to recognize the human's concurrently occurring composite tasks is a key requirement for realizing a successful HRT collaboration for unstructured and dynamic environments. This dissertation integrated a concurrency detection method into the task recognition algorithm to detect concurrent, composite tasks.

6.3 Future Work

The use of HRTs in unstructured and dynamic domains will not become feasible until humans can collaborate with robots as effectively as they do with other humans. Developing such a collaborative human-robot teaming architecture requires a robot to understand an individual's multi-dimensional task state and adapt to changing conditions accordingly. The developed multi-dimensional task recognition algorithm is a step toward realizing such human-robot collaboration. This research effort can be advanced to the next level by pursuing multiple future directions, many of which address the current algorithm's limitations. An overview of potential future research directions is presented in Table 6.2.

6.3.1 Adaptive Metric Segmentation

Most existing task recognition algorithms segment the sensor data into temporal chunks using a fixed window size that are the inputs to the machine learning algorithms. A short-duration task (e.g., Locate and Scan visual tasks) may require a smaller window size, so that the task is not overshadowed (e.g., confused) by unrelated data, while a long-duration task (e.g., Coordination visual task) may require a larger window size to provide sufficient context. Therefore, it may be necessary for a task recognition algorithm to use

Table 6.2: An Overview of the Future Research Directions

Future Research Directions
Adaptive Metric Segmentation
Out-of-class Task Recognition
Customized Recognition Models
Concurrent Atomic Task Detection
Modeling Task Transitions
Algorithmic Expansion to Detect Mission Tasks
Sensor Minimization Analysis
Real-Time Deployment Onboard a Robot

an adaptive sliding window approach [170, 194, 195]. This approach will permit expanding and contracting the window size based on the task, which may lead to more accurate detection. An ensemble learning algorithm may also be leveraged, where the algorithm makes predictions over multiple fixed window sizes and fuses the predictions across the window sizes intelligently to detect the tasks.

6.3.2 Out-of-Class Task Recognition

Due to the dynamic nature of certain task environments, humans will not always perform tasks that are known to the algorithm, which are called out-of-class tasks. Misclassification of an out-of-class task can result in a robot adapting its behavior incorrectly, causing more harm than good. Existing task recognition algorithms rarely detect anomalous instances and those that do require unknown negative examples during training for detecting out-of-class instances [142, 184], which may not be always available. An out-of-class task detection algorithm that can detect anomalous task instances automatically, without requiring training on negative task examples will be crucial for appropriate robot behavior adaptation in unstructured, dynamic domains [153].

6.3.3 Customized Recognition Models

Individual differences (e.g., strength levels, fatigue, training, expertise) result in humans performing the same task differently, often leading to different steps, step orderings, or completion times that can result in one task being mapped to multiple different sensor readings. Robots cannot adapt well in real-time due to these individual differences, which is a

foundational problem in collaborative HRTs that is exacerbated in unstructured, dynamic environments. Modeling these inconsistencies caused by individual human differences is challenging, but critical for future HRTs' mission success. Humans must train with their robot teammates, so that like humans, the robots can also develop customized models of their human teammates and adapt to changing conditions. While there have been some advances for addressing such individual differences, the existing approaches only considered gross motor and fine-grained motor tasks [9, 63, 164, 251], but not composite tasks involving multiple activity components. Transfer learning algorithms can be leveraged to customize the generalized task recognition algorithm to accommodate individual differences . Using such customized task recognition models can allow robots to autonomously adapt to their human teammates more accurately and effectively. The resulting system will enable improved team collaboration for complex domains (e.g., disaster response).

6.3.4 Concurrent Atomic Task Detection

The multi-dimensional task recognition algorithm assumed a task decomposition for which a human teammate can only be involved in one atomic task per activity component for any given instance. This assumption does not hold in general because an atomic task may actually involve multiple activity components (e.g., *Talking over a Walkie-Talkie* task requires tactile and speech components). Further, this assumption does not hold when the atomic tasks are influenced by extraneous factors independent of the human teammates. For instance, two or more auditory tasks may occur simultaneously (i.e., secondary prompts overlapping with the robot's sampling instructions). Therefore, algorithmic extensions that can accommodate such overlapping atomic tasks across components may be required.

6.3.5 Modeling Task Transitions

Inertial-based task recognition algorithms (e.g., gross motor, fine-grained motor, and tactile tasks) are vulnerable to signal variations, leading to challenges in detecting task transitions [35]. Modeling task transitions as a separate task group to indicate that a human teammate is transitioning from one task to the other can potentially mitigate this issue. Algorithms capable of detecting task transitions can be augmented with the existing multi-dimensional task recognition algorithm to inform task switching, so that appropriate robot behaviors can be adapted.

6.3.6 Algorithmic Expansion to Detect Mission Tasks

The existing multi-dimensional task recognition algorithm has a restricted scope that solely identifies atomic and composite tasks; however, accurately identifying the mission task is pivotal for pairing a human teammate with an appropriate robot counterpart. This scenario is particularly important for teams operating with heterogeneous robot teammates, where each robot teammate can be tailored to specific mission requirements. For instance, drone agents are well-suited for surveillance missions, whereas unmanned ground vehicles excel at terrestrial activities (e.g., clearing roadblocks and debris).

6.3.7 Sensor Minimization Analysis

HRTs will engage in a diverse set of tasks, involving differing combinations of activity components [20, 21]; therefore, adopting a multimodal approach that integrates metrics from various sensors is crucial. However, incorporating more than three metrics per activity element may lead to redundancy for some types of tasks and potentially hinder overall performance, as demonstrated in the supervisory-based evaluation. Thus, it is imperative to optimize input metrics to achieve the highest task recognition rate, while minimizing the number of multi-modal wearable sensors required for a wide range of tasks. Identifying the most pertinent metrics and sensors is essential to reducing the necessary number of wearable sensors. A sensor minimization and handedness analysis must be conducted for the peer-based evaluation in order to understand how certain design parameters (e.g., incorporated multimodal wearable metrics and handedness) affect the activity components' task recognition. The insights gained from this analysis will be invaluable in the development of task recognition algorithms for detecting HRT tasks in dynamic and uncertain environments.

6.3.8 Real-Time Deployment Onboard a Robot

The multi-dimensional task recognition algorithm's ability to detect tasks has only been validated post-hoc. A system capable of recognizing the tasks in real-time across all seven components has never been demonstrated. A real-time human teammate's task state estimation system that runs onboard a robot must be designed and deployed.

Bibliography

[1] EMOTIV - Brainwear® Wireless EEG Technology. http://emotiv.com/. Accessed: 2021-06-26.

[2] MUSE 2, InteraXon Inc., Muse™- Meditation Made Easy. https://choosemuse.com/muse-2/. Accessed: 2021-06-26.

[3] NeuroSky, Biosensors, NeuroSky Co. http://neurosky.com/. Accessed: 2021-06-26.

[4] M. Abdoli-Eramaki, C. Damecour, J. Christenson, and J. Stevenson. The effect of perspiration on the sEMG amplitude and power spectrum. *Journal of Electromyography and Kinesiology*, 22(6):908–913, 2012.

[5] J. Abdulbaqi. *Speech-based activity recognition for medical teamwork*. PhD thesis, Rutgers University-School of Graduate Studies, 2020.

[6] J. Abdulbaqi, Y. Gu, Z. Xu, C. Gao, I. Marsic, and R. S. Burd. Speech-based activity recognition for trauma resuscitation. In *IEEE International Conference on Healthcare Informatics*, pages 1–8. IEEE, 2020.

[7] U. Ahlstrom and F. J. Friedman-Berg. Using eye movement activity as a correlate of cognitive workload. *International Journal of Industrial Ergonomics*, 36(7):623–636, 2006.

[8] M. A. Al-qaness, A. Dahou, M. Abd Elaziz, and A. Helmi. Multi-ResAtt: Multilevel residual network with attention for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Informatics*, 19(1):144–152, 2022.

[9] L. Alawneh, M. Al-Ayyoub, Z. A. Al-Sharif, and A. Shatnawi. Personalized human activity recognition using deep learning and edge-cloud architecture. *Journal of Ambient Intelligence and Humanized Computing*, 14(9):12021–12033, 2023.

[10] H. Allahbakhshi, L. Conrow, B. Naimi, and R. Weibel. Using accelerometer and GPS data for real-life physical activity type detection. *Sensors*, 20(3):588, 2020.

[11] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[12] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan. Deep activity recognition models with triaxial accelerometers. In *Workshops at the AAAI Conference on Artificial Intelligence*, 2016.

[13] C. Amma, T. Krings, J. Böer, and T. Schultz. Advancing muscle-computer interfaces with high-density electromyography. In *ACM Conference on Human Factors in Computing Systems*, pages 929–938, 2015.

[14] M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed. Better physical activity classification using smartphone acceleration sensor. *Journal of Medical Systems*, 38(9):1–10, 2014.

[15] L. Atallah, B. Lo, R. King, and G.-Z. Yang. Sensor placement for activity detection using wearable accelerometers. In *IEEE International Conference on Body Sensor Networks*, pages 24–29. IEEE, 2010.

[16] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A. G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, 1(1):1–13, 2014.

[17] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic Convolutional and Recurrent Networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[18] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas. Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499, 2014.

[19] O. Banos, M. A. Toth, M. Damas, H. Pomares, and I. Rojas. Dealing with the effects of sensor displacement in wearable activity recognition. *Sensors*, 14(6):9995–10023, 2014.

[20] P. Baskaran and J. A. Adams. Multi-dimensional task recognition for human-robot teaming: Literature review. *Frontiers in Robotics and AI*, 10:1123374, 2023.

[21] P. Baskaran, J. B. Smith, and J. A. Adams. Visual task recognition for human-robot teams. In *IEEE International Conference on Human-Machine Systems*, pages 1–6. IEEE, 2022.

[22] I. Batzianoulis, S. El-Khoury, E. Pirondini, M. Coscia, S. Micera, and A. Billard. EMG-based decoding of grasp gestures in reaching-to-grasping motions. *Robotics and Autonomous Systems*, 91:59–70, 2017.

[23] M. Beck, T. Dyakowski, and R. Williams. Process tomography - the state of the art. *Transactions of the Institute of Measurement and Control*, 20(4):163–177, 1998.

[24] E. A. Bergs, F. L. Rutten, T. Tadros, P. Krijnen, and I. B. Schipper. Communication during trauma resuscitation: Do we know what is happening? *Injury*, 36(8):905–911, 2005.

[25] R. Biedert, J. Hees, A. Dengel, and G. Buscher. A robust realtime reading-skimming classifier. In *Eye Tracking Research and Applications Symposium*, pages 123–130, 2012.

[26] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, New York, USA, 2006.

[27] S. N. Boualia and N. E. B. Amara. Pose-based human activity recognition: A review. In *IEEE International Wireless Communications & Mobile Computing Conference*, pages 1468–1475. IEEE, 2019.

[28] R. Braojos, I. Beretta, J. Constantin, A. Burg, and D. Atienza. A wireless body sensor network for activity monitoring with low transmission overhead. In *IEEE International Conference on Embedded and Ubiquitous Computing*, pages 265–272. IEEE, 2014.

[29] W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3329–3336. IEEE, 2011.

[30] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster. Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753, 2010.

[31] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *IEEE International Joint Conference on Neural Networks*, pages 1–7. IEEE, 2015.

[32] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1291–1303, 2017.

[33] N. A. Capela, E. D. Lemaire, and N. Baddour. Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients. *PLoS One*, 10(4):e0124414, 2015.

[34] D. Castro, S. Hickson, V. Bettadapura, E. Thomaz, G. Abowd, H. Christensen, and I. Essa. Predicting daily activities from egocentric images using deep learning. In *ACM International Symposium on Wearable Computers*, pages 75–82, 2015.

[35] S. H. Cha, J. Seo, S. H. Baek, and C. Koo. Towards a well-planned, activity-based work environment: Automated recognition of office activities using accelerometers. *Building and Environment*, 144:86–93, 2018.

[36] S. Chakroborty, A. Roy, and G. Saha. Fusion of a complementary feature set with MFCC for improved closed set text-independent speaker identification. In *IEEE International Conference on Industrial Technology*, pages 387–390. IEEE, 2006.

[37] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[38] T. K. Chan and C. S. Chin. A Comprehensive review of polyphonic sound event detection. *IEEE Access*, 8:103339–103373, 2020.

[39] L. Chen and C. D. Nugent. *Human activity recognition and behaviour analysis.* Springer International Publishing, New York City, NY, 2019.

[40] L. Chen, X. Liu, L. Peng, and M. Wu. Deep learning based multimodal complex human activity recognition using wearable devices. *Applied Intelligence*, 51:4029–4042, 2021.

[41] X. Chen, X. Zhang, Z.-Y. Zhao, J.-H. Yang, V. Lantz, and K.-Q. Wang. Hand gesture recognition research based on surface EMG sensors and 2D-accelerometers. In *IEEE International Symposium on Wearable Computers*, pages 11–14. IEEE, 2007.

[42] X. Chen, X. Zhang, Z.-Y. Zhao, J.-H. Yang, V. Lantz, and K.-Q. Wang. Multiple hand gesture recognition based on surface EMG signal. In *IEEE International Conference on Bioinformatics and Biomedical Engineering*, pages 506–509. IEEE, 2007.

[43] Y. Chen and Y. Xue. A deep learning approach to human activity recognition based on single accelerometer. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 1488–1492. IEEE, 2015.

[44] P.-C. Chung and C.-D. Liu. A daily behavior enabled Hidden Markov Model for human behavior understanding. *Pattern Recognition*, 41(5):1572–1580, 2008.

[45] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay. Optimal placement of accelerometers for the detection of everyday activities. *Sensors*, 13(7):9183–9200, 2013.

[46] J. R. Comstock and R. J. Arnegard. The multi-attribute task battery for human operator workload and strategic behavior research. Technical report, 1992.

[47] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar. A Survey on activity detection and classification using wearable sensors. *IEEE Sensors*, 17(2):386–403, Jan 2017.

[48] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin. Deep learning for electromyographic hand gesture

signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.

[49] D. Dash, P. Ferrari, S. Malik, and J. Wang. Automatic speech activity recognition from MEG signals using seq2seq learning. In *IEEE International Conference on Engineering in Medicine and Biology Society International Conference on Neural Engineering*, pages 340–343. IEEE, 2019.

[50] S. Datta, A. Banerjee, A. Konar, and D. Tibarewala. Electrooculogram based cognitive context recognition. In *IEEE International Conference on Electronics, Communication and Instrumentation*, pages 1–4. IEEE, 2014.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, June 2009.

[52] H. Ding, L. Shangguan, Z. Yang, J. Han, Z. Zhou, P. Yang, W. Xi, and J. Zhao. FEMO: A platform for free-weight exercise monitoring with RFIDs. In *ACM Conference on Embedded Networked Sensor Systems*, pages 141–154, 2015.

[53] I. Dirgová Luptáková, M. Kubovčík, and J. Pospíchal. Wearable sensor-based human activity recognition with transformer model. *Sensors*, 22(5):1911, 2022.

[54] Y. Du, F. Chen, W. Xu, and Y. Li. Recognizing interaction activities using dynamic Bayesian network. In *IEEE International Conference on Pattern Recognition*, volume 1, pages 618–621. IEEE, 2006.

[55] M. Ermes, J. Parkka, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 4451–4454. IEEE, 2008.

[56] A. I. Faisal, S. Majumder, T. Mondal, D. Cowan, S. Naseh, and M. J. Deen. Monitoring methods of human body joints: State-of-the-art and research challenges. *Sensors*, 19(11), 2019.

[57] X. Fan, F. Wang, F. Wang, W. Gong, and J. Liu. When RFID meets deep learning: Exploring cognitive intelligence for activity identification. *IEEE Wireless Communications*, 26(3):19–25, 2019.

[58] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *IEEE International Conference on Computer Vision*, pages 407–414. IEEE, 2011.

[59] N. Fligge, H. Urbanek, and P. van der Smagt. Relation between object properties and EMG during reaching to grasp. *Journal of Electromyography and Kinesiology*, 23(2):402–410, 2013.

[60] D. Fortin-Simard, J.-S. Bilodeau, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane. Exploiting passive RFID technology for activity recognition in smart homes. *IEEE Intelligent Systems*, 30(4):7–15, 2015.

[61] J. Fortune, J. Heard, and J. A. Adams. Real-time speech workload estimation for intelligent human-machine systems. In *Human Factors and Ergonomics Society Annual Meeting*, volume 64, pages 334–338, 2020.

[62] A. E. Frank, A. Kubota, and L. D. Riek. Wearable activity recognition for robust human-robot teaming in safety-critical environments via hybrid neural networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 449–454, Nov 2019.

[63] Z. Fu, X. He, E. Wang, J. Huo, J. Huang, and D. Wu. Personalized human activity recognition based on integrated wearable sensor and transfer learning. *Sensors*, 21 (3):885, 2021.

[64] L. Gao, A. K. Bourke, and J. Nelson. A system for activity recognition using multi-sensor fusion. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 7869–7872. IEEE, 2011.

[65] Y. Gao, Y. Mishchenko, A. Shah, S. Matsoukas, and S. Vitaladevuni. Towards data-efficient modeling for wake word spotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7479–7483. IEEE, 2020.

[66] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–419, 2018.

[67] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 776–780. IEEE, 2017.

[68] H. Ghasemzadeh and R. Jafari. Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees. *IEEE Transactions on Industrial Informatics*, 7(1):66–77, 2010.

[69] A. N. Gilbert and C. J. Wysocki. Hand preference and age in the United States. *Neuropsychologia*, 30(7):601–608, 1992.

[70] H. Gjoreski, S. Kozina, M. Gams, and M. Luštrek. RAReFall—Real-time activity recognition and fall detection system. In *IEEE International Conference on Pervasive Computing and Communication Workshops*, pages 145–147. IEEE, 2014.

[71] H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams. Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer. In *International Joint Conference on Artificial Intelligence, Workshop on Deep Learning for Artificial Intelligence*, volume 10, page 970, 2016.

[72] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[73] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[74] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12): 2247–2253, 2007.

[75] M. Grana, M. Aguilar-Moreno, J. De Lope Asiain, I. B. Araquistain, and X. Garmendia. Improved activity recognition combining inertial motion sensors and electroencephalogram signals. *International Journal of Neural Systems*, 30(10):2050053, 2020.

[76] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[77] C. W. Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329–352, 1980.

[78] Y. Gu, R. Zhang, X. Zhao, S. Chen, J. Abdulbaqi, I. Marsic, M. Cheng, and R. S. Burd. Multimodal attention network for trauma activity recognition from spoken language and environmental sound. In *IEEE International Conference on Healthcare Informatics*, pages 1–6. IEEE, 2019.

[79] Y. Guo, D. Freer, F. Deligianni, and G.-Z. Yang. Eye-tracking for performance evaluation and workload estimation in space telerobotic training. *IEEE Transactions on Human-Machine Systems*, 52(1):1–11, 2021.

[80] C. E. Harriott, T. Zhang, and J. A. Adams. Assessing physical workload for human-robot peer-based teams. *International Journal of Human-Computer Studies*, 71(7-8): 821–837, 2013.

[81] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*, volume 2. Springer New York, NY, 2009.

[82] P. Haubrick and J. Ye. Robust audio sensing with multi-sound classification. In *IEEE International Conference on Pervasive Computing and Communications*, pages 1–7. IEEE, 2019.

[83] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[84] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[85] J. Heard, C. E. Harriott, and J. A. Adams. A survey of workload assessment algorithms. *IEEE Transactions on Human-Machine Systems*, 48(5):434–451, 2018.

[86] J. Heard, R. Heald, C. E. Harriott, and J. A. Adams. A diagnostic human workload assessment algorithm for human-robot teams. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 123–124, 2018.

[87] J. Heard, J. Fortune, and J. A. Adams. Speech workload estimation for human-machine interaction. In *Human Factors and Ergonomics Society Annual Meeting*, volume 63, pages 277–281, 2019.

[88] J. Heard, R. A. Paris, D. Scully, C. McNaughton, J. M. Ehrenfeld, J. Coco, D. Fabbri, B. Bodenheimer, and J. A. Adams. Automatic clinical procedure detection for emergency services. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 337–340, July 2019.

[89] J. R. Heard. *An adaptive supervisory-based human-robot teaming architecture*. PhD thesis, Vanderbilt University, 2019.

[90] R. P. Henderson and J. G. Webster. An impedance camera for spatially specific measurements of the thorax. *IEEE Transactions on Biomedical Engineering*, BME-25(3):250–254, May 1978.

[91] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 131–135. IEEE, 2017.

[92] P. Hevesi, J. A. Ward, O. Amiraslanov, G. Pirkl, and P. Lukowicz. Wearable eye tracking for multisensor physical activity recognition. *International Journal on Advances in Intelligent Systems*, 10(1-2):103–116, 2018.

[93] C.-P. Hsiao, R. Li, X. Yan, and E. Y.-L. Do. Tactile teacher: Sensing finger tapping in piano playing. In *ACM International Conference on Tangible, Embedded, and Embodied Interaction*, pages 257–260, 2015.

[94] G. Hu, X. Qiu, and L. Meng. RTagCare: Deep human activity recognition powered by passive computational RFID sensors. In *IEEE Asia-Pacific Network Operations and Management Symposium*, pages 1–4. IEEE, 2016.

[95] A. Ignatov. Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Applied Soft Computing*, 62:915–922, 2018.

[96] M. Inoue, S. Inoue, and T. Nishida. Deep Recurrent Neural Network for mobile human activity recognition with high throughput. *Artificial Life and Robotics*, 23 (2):173–185, 2018.

[97] S. Ishimaru, K. Kunze, K. Kise, J. Weppner, A. Dengel, P. Lukowicz, and A. Bulling. In the blink of an eye: Combining head motion and eye blink frequency for activity recognition with Google Glass. In *ACM Augmented Human International Conference*, pages 1–4. ACM, 2014.

[98] S. Ishimaru, K. Kunze, Y. Uema, K. Kise, M. Inami, and K. Tanaka. Smarter eyewear: Using commercial EOG glasses for activity recognition. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 239–242, 2014.

[99] S. Ishimaru, K. Hoshika, K. Kunze, K. Kise, and A. Dengel. Towards reading trackers in the wild: Detecting reading activities by EOG glasses and deep neural networks. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, pages 704–711, 2017.

[100] M. R. Islam, S. Sakamoto, Y. Yamada, A. W. Vargo, M. Iwata, M. Iwamura, and K. Kise. Self-supervised learning for reading activity classification. *ACM Transactions on Interactive, Mobile, Wearable Ubiquitous Technologies*, 5(3), sep 2021.

[101] T. Iwamoto and H. Shinoda. Finger ring device for tactile sensing and human machine interface. In *IEEE Annual Conference of Society of Instrument and Control Engineers*, pages 2132–2136. IEEE, 2007.

[102] S. Jagannath, A. Sarcevic, and I. Marsic. An analysis of speech as a modality for activity recognition during complex medical teamwork. In *ACM/EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 88–97, 2018.

[103] S. Jagannath, A. Sarcevic, N. Kamireddi, and I. Marsic. Assessing the feasibility of speech-based activity recognition in dynamic medical settings. In *ACM Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[104] A. Jalal, S. Kamal, and D. Kim. A depth video-based human detection and activity recognition using multi-features and embedded Hidden Markov Models for health care monitoring systems. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4(4), 2017.

[105] A. Jalal, M. A. K. Quaid, S. B. u. d. Tahir, and K. Kim. A study of accelerometer and gyroscope measurements in physical life-log activities detection systems. *Sensors*, 20 (22), 2020.

[106] M. Janidarmian, A. Roshan Fekr, K. Radecka, and Z. Zilic. A comprehensive analysis on wearable acceleration sensors in human activity recognition. *Sensors*, 17(3):529, 2017.

[107] R. Jia and B. Liu. Human daily activity recognition by fusing accelerometer and multi-lead ECG data. In *IEEE International Conference on Signal Processing, Communication and Computing*, pages 1–4. IEEE, 2013.

[108] S. Jiang, Y. Cao, S. Iyengar, P. Kuryloski, R. Jafari, Y. Xue, R. Bajcsy, and S. Wicker. CareNet: An integrated wireless sensor networking environment for remote healthcare. In *ICST International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[109] L. Jing, Y. Zhou, Z. Cheng, and J. Wang. A recognition method for one-stroke finger gestures using a MEMS 3D accelerometer. *Institute of Electronics, Information and Communication Engineers Transactions on Information and Systems*, 94(5):1062–1072, 2011.

[110] A. Jordao, A. C. Nazare, J. Sena, and W. R. Schwartz. Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. *arXiv*, 2018.

[111] Z. Ju and H. Liu. Human hand motion analysis with multisensory information. *IEEE/ASME Transactions on Mechatronics*, 19(2):456–466, April 2014.

[112] P. Kaczmarek, T. Mańkowski, and J. Tomczyński. putEMG — A surface electromyography hand gesture recognition dataset. *Sensors*, 19(16), 2019.

[113] A. Kawazoe, M. Di Luca, and Y. Visell. Tactile Echoes: A wearable system for tactile augmentation of objects. In *IEEE World Haptics Conference*, pages 359–364. IEEE, 2019.

[114] S.-R. Ke, H. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi. A review on video-based human activity recognition. *Computers*, 2(2):88–131, 2013.

[115] J. F. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems*, 2(1): 26–41, 1984.

[116] C. Kelton, Z. Wei, S. Ahn, A. Balasubramanian, S. R. Das, D. Samaras, and G. Zelinsky. Reading detection in real-time. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–5, 2019.

[117] L. Kester and P. A. Kirschner. *Cognitive tasks and learning*, pages 619–622. Springer US, Boston, MA, 2012.

[118] U. M. Khan, Z. Kabir, S. A. Hassan, and S. H. Ahmed. A Deep learning framework using passive WiFi sensing for respiration monitoring. In *IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[119] R. Kher, T. Pawar, and V. Thakar. Combining accelerometer data with Gabor energy feature vectors for body movements classification in ambulatory ECG signals. In *IEEE International Conference on Biomedical Engineering and Informatics*, pages 413–417. IEEE, 2013.

[120] P. Kiefer, I. Giannopoulos, and M. Raubal. Using eye movements to recognize activities on cartographic maps. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 488–491, 2013.

[121] Y. Kim and B. Toomajian. Hand gesture recognition using micro-doppler signatures with Convolutional Neural Network. *IEEE Access*, 4:7125–7130, 2016.

[122] Y.-J. Kim, B.-N. Kang, and D. Kim. Hidden Markov Model ensemble for activity recognition using tri-axis accelerometer. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 3036–3041. IEEE, 2015.

[123] D. P. Kingma and J. Ba. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[124] W. Klimesch. EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29(2-3):169–195, 1999.

[125] M. H. Kolekar and D. P. Dash. Hidden Markov Model based human activity recognition using shape and optical flow based features. In *IEEE Region 10 Conference*, pages 393–397. IEEE, 2016.

[126] S. Kollmorgen and K. Holmqvist. Automatically detecting reading in eye tracking data. *Lund University Cognitive Studies*, pages 1–9, 2007.

[127] H. Koskimaki, V. Huikari, P. Siirtola, P. Laurinen, and J. Roning. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *IEEE Mediterranean Conference on Control and Automation*, pages 401–405. IEEE, 2009.

[128] H. Koskimäki, P. Siirtola, and J. Röning. Myogym: introducing an open gym data set for activity recognition collected using Myo armband. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, pages 537–546, 2017.

[129] N. C. Krishnan and S. Panchanathan. Analysis of low resolution accelerometer data for continuous human activity recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3337–3340. IEEE, 2008.

[130] A. Kubota, T. Iqbal, J. A. Shah, and L. D. Riek. Activity recognition in manufacturing: The roles of motion capture and sEMG+inertial wearables in detecting fine vs. gross motion. In *IEEE International Conference on Robotics and Automation*, pages 6533–6539, 2019.

[131] S. S. Kumar and M. John. Human activity recognition using optical flow based feature set. In *IEEE International Carnahan Conference on Security Technology*, pages 1–5. IEEE, 2016.

[132] K. Kunze, H. Kawaichi, K. Yoshimura, and K. Kise. Towards inferring language expertise using eye tracking. In *ACM Conference on Human Factors in Computing Systems*, pages 217–222. 2013.

[133] K. Kunze, Y. Shiga, S. Ishimaru, and K. Kise. Reading activity recognition using an off-the-shelf EEG – detecting reading activities and distinguishing genres of documents. In *IEEE International Conference on Document Analysis and Recognition*, pages 96–100, 2013.

[134] K. Kunze, Y. Utsumi, Y. Shiga, K. Kise, and A. Bulling. I know what you are reading: Recognition of document types using mobile eye tracking. In *ACM International Symposium on Wearable Computers*, pages 113–116, 2013.

[135] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[136] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22(1):949–961, 2019.

[137] A. Ladjailia, I. Bouchrika, H. F. Merouani, N. Harrati, and Z. Mahfouf. Human activity recognition via optical flow: Decomposing activities into basic actions. *Neural Computing and Applications*, 32(21):16387–16400, 2020.

[138] P. Lagodzinski, K. Shirahama, and M. Grzegorzek. Codebook-based electrooculography data analysis towards cognitive activity recognition. *Computers in Biology and Medicine*, 95:277–287, 2018.

[139] G. Lan, B. Heit, T. Scargill, and M. Gorlatova. GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior. In *Embedded Networked Sensor Systems*, pages 422–435, 2020.

[140] M. Landsmann, O. Augereau, and K. Kise. Classification of reading and not reading behavior based on eye movement analysis. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing and ACM International Symposium on Wearable Computers*, pages 109–112, 2019.

[141] N. D. Lane, P. Georgiev, and L. Qendro. DeepEar: Robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, page 283–294, New York, NY, USA, 2015. Association for Computing Machinery.

[142] G. Laput and C. Harrison. Sensing fine-grained hand activity with smartwatches. In *ACM Conference on Human Factors in Computing Systems*, pages 1—-13, New York, NY, USA, 2019. Association for Computing Machinery.

[143] G. Laput, C. Yang, R. Xiao, A. Sample, and C. Harrison. EM-Sense: Touch recognition of uninstrumented, electrical and electromechanical objects. In *ACM Symposium on User Interface Software and Technology*, page 157–166, New York, NY, USA, 2015. Association for Computing Machinery.

[144] G. Laput, K. Ahuja, M. Goel, and C. Harrison. Ubicoustics: Plug-and-play acoustic activity recognition. In *ACM Symposium on User Interface Software and Technology*, pages 213–224, 2018.

[145] O. D. Lara and M. A. Labrador. A mobile platform for real-time human activity recognition. In *IEEE Consumer Communications and Networking Conference*, pages 667–671. IEEE, 2012.

[146] O. D. Lara and M. A. Labrador. A Survey on human activity recognition using wearable sensors. *IEEE Communications Surveys Tutorials*, 15(3):1192–1209, 2013.

[147] O. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, 8(5):717–729, 2012.

[148] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal Convolutional Networks for action segmentation and detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[149] K.-F. Lee, H.-W. Hon, and R. Reddy. An overview of the SPHINX speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):35–45, 1990.

[150] M. H. Lee and H. R. Nicholls. Review article tactile sensing for mechatronics — a state of the art survey. *Mechatronics*, 9(1):1–31, 1999.

[151] S.-M. Lee, S. M. Yoon, and H. Cho. Human activity recognition from accelerometer data using Convolutional Neural Network. In *IEEE International Conference on Big Data and Smart Computing*, pages 131–134. IEEE, 2017.

[152] Y.-S. Lee and S.-B. Cho. Activity recognition using hierarchical Hidden Markov Models on a smartphone with 3D accelerometer. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 460–467. Springer, 2011.

[153] J. Li, H. Xu, and Y. Wang. Multi-resolution Fusion Convolutional Network for Open Set Human Activity Recognition. *IEEE Internet of Things Journal*, 10(13):11369–11382, 2023.

[154] L. Li, R. A. Paris, C. Pinson, Y. Wang, J. Coco, J. Heard, J. A. Adams, D. V. Fabbri, and B. Bodenheimer. Emergency clinical procedure detection with deep learning. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 158–163. IEEE, 2020.

[155] M. Li, V. Rozgić, G. Thatte, S. Lee, A. Emken, M. Annavaram, U. Mitra, D. Spruijt-Metz, and S. Narayanan. Multimodal physical activity recognition by fusing temporal and cepstral information. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(4):369–380, 2010.

[156] X. Li, D. Yao, X. Pan, J. Johannaman, J. Yang, R. Webman, A. Sarcevic, I. Marsic, and R. S. Burd. Activity recognition for medical teamwork based on passive RFID. In *IEEE International Conference on RFID*, pages 1–9. IEEE, 2016.

[157] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd. Deep learning for RFID-based activity recognition. In *ACM Conference on Embedded Network Sensor Systems*, pages 164–175, 2016.

[158] X. Li, J. Luo, and R. Younes. ActivityGAN: Generative adversarial networks for data augmentation in sensor-based human activity recognition. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 249–254, 2020.

[159] D. Liang and E. Thomaz. Audio-based activities of daily living recognition with large-scale acoustic embeddings from online videos. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–18, 2019.

[160] D. Liang, G. Li, R. Adaimi, R. Marculescu, and E. Thomaz. Audioimu: Enhancing inertial sensing-based activity recognition with acoustic models. In *ACM International Symposium on Wearable Computers*, pages 44–48, 2022.

[161] J. Liao, J. Hu, and L. Liu. Recognizing complex activities by a temporal causal network-based model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 341–357. Springer, 2020.

[162] L. Liao, D. Fox, and H. Kautz. Hierarchical Conditional Random Fields for GPS-based activity recognition. In *Robotics Research*, pages 487–506. Springer, 2007.

[163] I. Lillo, J. C. Niebles, and A. Soto. Sparse composition of body poses and atomic actions for human activity recognition in RGB-D videos. *Image and Vision Computing*, 59:63–75, 2017.

[164] C.-Y. Lin and R. Marculescu. Model Personalization for Human Activity Recognition. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 1–7, 2020.

[165] L. Liu, S. Wang, G. Su, Z.-G. Huang, and M. Liu. Towards complex activity recognition using a Bayesian network-based probabilistic generative framework. *Pattern Recognition*, 68:295–309, 2017.

[166] X. Liu, S. Rajan, N. Ramasarma, P. Bonato, and S. I. Lee. The use of a finger-worn accelerometer for monitoring of hand use in ambulatory settings. *IEEE Journal of Biomedical and Health Informatics*, 23(2):599–606, 2018.

[167] L. E. Locascio, B. Harper, M. Robinson, and T. Badar. Standard practice for bulk sample collection and swab sample collection of visible powders suspected of being biological agents from nonporous surfaces: collaborative study. *Journal of AOAC International*, 90(1):299–333, 2007.

[168] Y. Lu, C. Zhang, B.-Y. Zhou, X.-P. Gao, and Z. Lv. A dual model approach to EOG-based human activity recognition. *Biomedical Signal Processing and Control*, 45:50–57, 2018.

[169] Y. Luo, Y. Li, P. Sharma, W. Shou, K. Wu, M. Foshey, B. Li, T. Palacios, A. Torralba, and W. Matusik. Learning human-environment interactions using conformal tactile textiles. *Nature Electronics*, 4(3):193–201, 2021.

[170] C. Ma, W. Li, J. Cao, J. Du, Q. Li, and R. Gravina. Adaptive sliding window based activity recognition for assisted livings. *Information Fusion*, 53:55–65, 2020.

[171] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1903, 2016.

[172] A. Mannini and A. M. Sabatini. On-line classification of human activity and estimation of walk-run speed from acceleration data using support vector machines. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 3302–3305. IEEE, 2011.

[173] G. Marquart, C. Cabrall, and J. de Winter. Review of eye-related measures of drivers' mental workload. *Procedia Manufacturing*, 3:2854–2861, 2015.

[174] F. Martinez, E. Pissaloux, and A. Carbone. Towards activity recognition from eye-movements using contextual temporal learning. *Integrated Computer-Aided Engineering*, 24(1):1–16, 2017.

[175] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez. Metric learning for novelty and anomaly detection. *arXiv preprint arXiv:1808.05492*, 2018.

[176] K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 565–570, June 2014.

[177] W. W. Mayol and D. W. Murray. Wearable hand activity recognition for event summarization. In *IEEE International Symposium on Wearable Computers*, pages 122–129, Oct 2005.

[178] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul. Deep residual networks for human activity recognition based on biosignals from wearable devices. In *IEEE International Conference on Telecommunications and Signal Processing*, pages 310–313. IEEE, 2022.

[179] S. Mekruksavanich, A. Jitpattanakul, K. Sitthithakerngkiet, P. Youplao, and P. Yupapin. Resnet-SE: Channel attention-based deep residual network for complex activity recognition using wrist-worn wearable sensors. *IEEE Access*, 10:51142–51154, 2022.

[180] N. Mennie, M. Hayhoe, and B. Sullivan. Look-ahead fixations: Anticipatory eye movements in natural tasks. *Experimental Brain Research*, 179(3):427–442, 2007.

[181] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen. DCASE 2017 challenge setup: Tasks, datasets and baseline system. In *Detection and Classification of Acoustic Scenes and Events Workshops*, 2017.

[182] J. Meyer, A. Frank, T. Schlebusch, and E. Kasneci. U-HAR: A convolutional approach to human activity recognition combining head and eye movements for context-aware smart glasses. *ACM on Human-Computer Interaction*, 6(ETRA):1–19, 2022.

[183] C. Min, N. F. Ince, and A. H. Tewfik. Generalization capability of a wearable early morning activity detection system. In *IEEE European Signal Processing Conference*, pages 1556–1560, Sep. 2007.

[184] C.-H. Min, N. F. Ince, and A. H. Tewfik. Early morning activity detection using acoustics and wearable wireless sensors. In *IEEE European Signal Processing Conference*, pages 1–5. IEEE, 2008.

[185] J. Min and S. Cho. Activity recognition based on wearable sensors using selection/fusion hybrid ensemble. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 1319–1324, Oct 2011.

[186] D. K. Mitchell. Mental workload and ARL workload modeling tools. Technical report, Army Research Lab, Aberdeen Proving Ground, MD, 2000.

[187] V. Mollyn, K. Ahuja, D. Verma, C. Harrison, and M. Goel. SAMoSA: Sensing Activities with Motion and Subsampled Audio. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–19, 2022.

[188] J. Mostow, K.-M. Chang, and J. Nelson. Toward exploiting EEG input in a reading tutor. In *International Conference on Artificial Intelligence in Education*, pages 230–237. Springer, 2011.

[189] T. S. Motwani and R. J. Mooney. Improving video activity recognition using object recognition and text mining. In *European Conference on Artificial Intelligence*, page 600–605, NLD, 2012. IOS Press.

[190] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, page 807–814, Madison, WI, USA, 2010. Omnipress.

[191] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering*, 50(6):711–723, 2003.

[192] A. Nandy, J. Saha, and C. Chowdhury. Novel features for intensive human activity recognition based on wearable and smartphone sensors. *Microsystem Technologies*, 26:1889—-1903, 2020.

[193] S. Neili Boualia and N. Essoukri Ben Amara. Deep full-body HPE for activity recognition from RGB frames only. *Informatics*, 8(1), 2021.

[194] Q. Ni, T. Patterson, I. Cleland, and C. Nugent. Dynamic detection of window starting positions and its implementation within an activity recognition framework. *Journal of Biomedical Informatics*, 62:171–180, 2016.

[195] M. H. M. Noor, Z. Salcic, I. Kevin, and K. Wang. Adaptive sliding window segmentation for physical activity recognition using a single tri-axial accelerometer. *Pervasive and Mobile Computing*, 38:41–59, 2017.

[196] G. A. Oguntala, R. A. Abd-Alhameed, N. T. Ali, Y.-F. Hu, J. M. Noras, N. N. Eya, I. Elfergani, and J. Rodriguez. SmartWall: Novel RFID-enabled ambient human activity recognition using machine learning for unobtrusive health monitoring. *IEEE Access*, 7:68022–68033, 2019.

[197] P. Olsson. Real-time and offline filters for eye tracking. Master's degree project, KTH Electrical Engineering, Stockholm, Sweden, 2007.

[198] R. T. Olszewski. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD thesis, 2001.

[199] F. J. Ordóñez and D. Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016.

[200] C. Orr, C. Nugent, H. Wang, and H. Zheng. A multi agent approach to facilitate the identification of interleaved activities. In *ACM International Conference on Digital Health*, pages 126–130, 2018.

[201] O. Ozioko, W. Taube, M. Hersh, and R. Dahiya. SmartFingerBraille: A tactile sensing and actuation based communication glove for deafblind people. In *IEEE International Symposium on Industrial Electronics*, pages 2014–2018. IEEE, 2017.

[202] S. Pancholi and R. Agarwal. Development of low cost EMG data acquisition system for Arm Activities Recognition. In *IEEE International Conference on Advances in Computing, Communications and Informatics*, pages 2465–2469. IEEE, 2016.

[203] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54(2):1–38, 2021.

[204] G. Parascandolo, H. Huttunen, and T. Virtanen. Recurrent neural networks for polyphonic sound event detection in real life recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6440–6444. IEEE, 2016.

[205] H. Park, S.-Y. Dong, M. Lee, and I. Youn. The role of heart-rate variability parameters in activity recognition and energy-expenditure estimation using wearable sensors. *Sensors*, 17(7):1698, 2017.

[206] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):119–128, 2006.

[207] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci. Xsens MTw Awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3D kinematic applications. *Xsens: Enschede, The Netherlands*, pages 1–9, 2018.

[208] T. Pawar, S. Chaudhuri, and S. P. Duttagupta. Analysis of ambulatory ECG signal. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 3094–3097. IEEE, 2006.

[209] T. Pawar, N. Anantakrishnan, S. Chaudhuri, and S. P. Duttagupta. Transition detection in body movement activities for wearable ECG. *IEEE Transactions on Biomedical Engineering*, 54(6):1149–1152, 2007.

[210] T. Pawar, N. S. Anantakrishnan, S. Chaudhuri, and S. P. Duttagupta. Impact analysis of body movement in ambulatory ECG. In *IEEE International Conference on Engineering in Medicine and Biology Society*, pages 5453–5456, 2007.

[211] J. B. Pelz, R. Canosa, and J. Babcock. Extended tasks elicit complex eye movement patterns. In *ACM Symposium on Eye Tracking Research and Applications*, pages 37–43, 2000.

[212] L. Peng, L. Chen, Z. Ye, and Y. Zhang. Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):1–16, 2018.

[213] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[214] P. Perera and V. M. Patel. Deep transfer learning for multiple class novelty detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11544–11552, 2019.

[215] M. Peters, S. Reimers, and J. T. Manning. Hand preference for writing and associations with selected demographic and behavioral variables in 255,100 subjects: the BBC internet study. *Brain and Cognition*, 62(2):177–189, 2006.

[216] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2847–2854. IEEE, 2012.

[217] G. Porter, T. Troscianko, and I. D. Gilchrist. Effort during visual search and counting: Insights from pupillometry. *Quarterly Journal of Experimental Psychology*, 60(2): 211–229, 2007.

[218] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesel. The Kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 320–324. IEEE, 2011.

[219] R. Pramanik, R. Sikdar, and R. Sarkar. Transformer-based deep reverse attention network for multi-sensory human activity recognition. *Engineering Applications of Artificial Intelligence*, 122:106150, 2023.

[220] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert. Wav2letter++: A fast open-source speech recognition system. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6460–6464. IEEE, 2019.

[221] C. M. Privitera, L. W. Renninger, T. Carney, S. Klein, and M. Aguilar. Pupil dilation during visual target detection. *Journal of Vision*, 10(10):1–14, 2010.

[222] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67, May 2016.

[223] E. Rahimian, S. Zabihi, A. Asif, and A. Mohammadi. Hybrid deep Neural Networks for sparse surface EMG-based hand gesture recognition. In *IEEE Asilomar Conference on Signals, Systems, and Computers*, pages 371–374. IEEE, 2020.

[224] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *AAAI Conference on Innovative Applications of Artificial Intelligence*, page 1541–1546. AAAI Press, 2005.

[225] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, 6(2):1–27, 2010.

[226] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[227] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *IEEE International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.

[228] D. Riboni and C. Bettini. COSAR: Hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[229] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millàn. Collecting complex activity datasets in highly rich networked sensor environments. In *IEEE International Conference on Networked Sensing Systems*, pages 233–240. IEEE, 2010.

[230] Y. Saez, A. Baldominos, and P. Isasi. A comparison study of classifier algorithms for cross-person physical activity recognition. *Sensors*, 17(1):66, 2017.

[231] M. Safyan, Z. U. Qayyum, S. Sarwar, R. García-Castro, and M. Ahmed. Ontology-driven semantic unified modeling for concurrent activity recognition (OSCAR). *Multimedia Tools and Applications*, 78(2):2073–2104, 2019.

[232] S. Saguna, A. Zaslavsky, and D. Chakraborty. Complex activity recognition using context-driven activity theory and activity signatures. *ACM Transactions on Computer-Human Interaction*, 20(6):1–34, 2013.

[233] J. Salamon and J. P. Bello. Deep Convolutional Neural Networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3): 279–283, 2017.

[234] A. Salehzadeh, A. P. Calitz, and J. Greyling. Human activity recognition using deep electroencephalography learning. *Biomedical Signal Processing and Control*, 62:102094, 2020.

[235] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[236] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *ACM Symposium on Eye Tracking Research and Applications*, pages 71–78. ACM, 2000.

[237] Y. Santiago-Espada, R. R. Myer, K. A. Latorella, and J. R. Comstock Jr. The multi-attribute task battery II (MATB-II) software for human performance and workload research: A user's guide (NASA/TM-2011–217164). *Hampton, VA: National Aeronautics and Space Administration, Langley Research Center*, 2011.

[238] S. Sarkar, K. Reddy, A. Dorgan, C. Fidopiastis, and M. Giering. Wearable EEG-based activity recognition in PHM-related service environment via deep learning. *International Journal of Prognostics and Health Management*, 7:1–10, 2016.

[239] M. Sathiyanarayanan and S. Rajan. Myo Armband for physiotherapy healthcare: A case study using gesture recognition application. In *IEEE International Conference on Communication Systems and Networks*, pages 1–6. IEEE, 2016.

[240] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[241] E. Scheme and K. Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.

[242] M. Scherhäufl, M. Pichler, and A. Stelzer. UHF RFID localization based on phase evaluation of passive tag arrays. *IEEE Transactions on Instrumentation and Measurement*, 64(4):913–922, 2014.

[243] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017.

[244] M. A. Schmuckler. What is ecological validity? A dimensional analysis. *Infancy*, 2 (4):419–436, 2001.

[245] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.

[246] S. Scrafton, M. J. Stainer, and B. W. Tatler. Coordinating vision and action in natural behaviour: Differences in spatiotemporal coupling in everyday tasks. *Canadian Journal of Experimental Psychology*, 71(2):133–145, 2017.

[247] X. Sha, C. Lian, Y. Zhao, J. Yu, S. Wang, and W. J. Li. An explicable keystroke recognition algorithm for customizable ring-type keyboards. *IEEE Access*, 8:22933–22944, 2020.

[248] S. R. Shakya, C. Zhang, and Z. Zhou. Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *International Journal of Machine Learning and Computing*, 8(6):577–582, 2018.

[249] K. Simonyan and A. Zisserman. Very deep Convolutional Networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[250] J. B. Smith, P. Baskaran, and J. A. Adams. Decomposed Physical Workload Estimation for Human-Robot Teams. In *IEEE International Conference on Human-Machine Systems*, pages 1–6. IEEE, 2022.

[251] E. Soleimani and E. Nazerfard. Cross-subject transfer learning in human activity recognition systems using generative adversarial networks. *Neurocomputing*, 426: 26–34, 2021.

[252] M. S. Sorower. A literature survey on algorithms for multi-label learning. Technical report, Oregon State University, 2010.

[253] N. Srivastava, J. Newn, and E. Velloso. Combining low and mid-level gaze features for desktop activity recognition. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–27, 2018.

[254] J. Steil and A. Bulling. Discovery of everyday human activities from long-term visual behaviour using topic models. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 75–85, 2015.

[255] C. Steinberg, F. Philippon, M. Sanchez, P. Fortier-Poisson, G. O'Hara, F. Molin, J.-F. Sarrazin, I. Nault, L. Blier, K. Roy, B. Plourde, and J. Champagne. A novel wearable device for continuous ambulatory ECG recording: Proof of concept and assessment of signal quality. *Biosensors*, 9(1), 2019.

[256] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras. Audio-based human activity recognition using non-markovian ensemble voting. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 509–514. IEEE, 2012.

[257] Y. Tang, J. Lu, Z. Wang, M. Yang, and J. Zhou. Learning semantics-preserving attention and contextual interaction for group activity recognition. *IEEE Transactions on Image Processing*, 28(10):4997–5012, 2019.

[258] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *IEEE International Symposium on Wearable Computers*, pages 37–40. IEEE, 2007.

[259] N. T. H. Thu and D. S. Han. HiHAR: A hierarchical hybrid deep learning architecture for wearable sensor-based human activity recognition. *IEEE Access*, 9: 145271–145281, 2021.

[260] D. Triboan, L. Chen, F. Chen, and Z. Wang. Semantic segmentation of real-time sensor data stream for complex activity recognition. *Personal and Ubiquitous Computing*, 21(3):411–425, 2017.

[261] E. Trigili, L. Grazi, S. Crea, A. Accogli, J. Carpaneto, S. Micera, N. Vitiello, and A. Panarese. Detection of movement onset using EMG signals for upper-limb exoskeletons in reaching tasks. *Journal of Neuroengineering and Rehabilitation*, 16(1): 1–16, 2019.

[262] T.-H. Tsai and P.-C. Hao. Customized wake-up word with keyword spotting using convolutional neural network. In *IEEE International SoC Design Conference*, pages 136–137. IEEE, 2019.

[263] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[264] A. Ullah, K. Muhammad, J. Del Ser, S. W. Baik, and V. H. C. de Albuquerque. Activity recognition using temporal optical flow convolutional features and multilayer LSTM. *IEEE Transactions on Industrial Electronics*, 66(12):9692–9702, 2018.

[265] D. L. Vail, M. M. Veloso, and J. D. Lafferty. Conditional random fields for activity recognition. In *ACM International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8, 2007.

[266] P. Vepakomma, D. De, S. K. Das, and S. Bhansali. A-Wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities. In *IEEE International Conference on Wearable and Implantable Body Sensor Networks*, pages 1–6, June 2015.

[267] B. Wahn, D. P. Ferris, W. D. Hairston, and P. König. Pupil sizes scale with attentional load and task experience in a multiple object tracking task. *PLoS One*, 11 (12):1–15, 12 2016.

[268] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu. Human activity recognition in a smart home environment with stacked denoising Autoencoders. In S. Song and Y. Tong, editors, *Web-Age Information Management*, pages 29–40, Cham, 2016. Springer International Publishing.

[269] E. J. Wang, T.-J. Lee, A. Mariakakis, M. Goel, S. Gupta, and S. N. Patel. MagnifiSense: Inferring device interaction using wrist-worn passive magneto-inductive sensors. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, page 15–26, New York, NY, USA, 2015. Association for Computing Machinery.

[270] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan. SensoryGANs: An effective generative adversarial framework for sensor-based human activity recognition. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2018.

[271] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity Recognition: A survey. *Pattern Recognition Letters*, 119:3–11, Mar. 2019.

[272] Z. Wang, M. Jiang, Y. Hu, and H. Li. An incremental learning method based on probabilistic Neural Networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16(4):691–699, 2012.

[273] S. Weng, L. Xiang, W. Tang, H. Yang, L. Zheng, H. Lu, and H. Zheng. A low power and high accuracy MEMS sensor-based activity recognition algorithm. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 33–38. IEEE, 2014.

[274] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[275] W. H. Wu, A. A. Bui, M. A. Batalin, L. K. Au, J. D. Binney, and W. J. Kaiser. MEDIC: Medical embedded device for individualized care. *Artificial Intelligence in Medicine*, 42(2):137–152, 2008.

[276] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan. InnoHAR: A deep Neural Network for complex human activity recognition. *IEEE Access*, 7:9893–9902, 2019.

[277] W. Xu, M. Zhang, A. A. Sawchuk, and M. Sarrafzadeh. Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks. In *IEEE International Conference on Wearable and Implantable Body Sensor Networks*, pages 124–129. IEEE, 2012.

[278] Y. Yan, E. Ricci, G. Liu, and N. Sebe. Egocentric daily activity recognition via multitask clustering. *IEEE Transactions on Image Processing*, 24(10):2984–2995, Oct 2015.

[279] K. Yatani and K. N. Truong. Bodyscope: a wearable acoustic sensor for activity recognition. In *ACM Conference on Ubiquitous Computing*, pages 341–350, 2012.

[280] F. Yu, V. Koltun, and T. Funkhouser. Dilated Residual Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 472–480, 2017.

[281] Y. Yuan, K.-m. Chang, J. N. Taylor, and J. Mostow. Toward unobtrusive measurement of reading comprehension using low-cost EEG. In *ACM International Conference on Learning Analytics and Knowledge*, pages 54–58, 2014.

[282] S. Zhang, M. Ang, W. Xiao, and C. Tham. Detection of activities for daily life surveillance: Eating and drinking. In *IEEE International Conference on eHealth Networking, Applications and Services*, pages 171–176. IEEE, 2008.

[283] S. Zhang, P. McCullagh, C. Nugent, and H. Zheng. Activity monitoring using a smartphone's accelerometer with hierarchical classification. In *IEEE International Conference on Intelligent Environments*, pages 158–163. IEEE, 2010.

[284] X. Zhang, X. Chen, W.-h. Wang, J.-h. Yang, V. Lantz, and K.-q. Wang. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *ACM International Conference on Intelligent User Interfaces*, pages 401–406, 2009.

[285] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang. A framework for hand gesture recognition based on accelerometer and EMG sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(6):1064–1076, 2011.

[286] X. Zhang, L. Yao, X. Wang, W. Zhang, S. Zhang, and Y. Liu. Know your mind: Adaptive cognitive activity recognition with reinforced CNN. In *IEEE International Conference on Data Mining*, pages 896–905. IEEE, 2019.

[287] Y. Zhang and C. Harrison. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *ACM Symposium on User Interface Software Technology*, page 167–173, New York, NY, USA, 2015. Association for Computing Machinery.

[288] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, and Q. Ji. Modeling temporal interactions with interval temporal Bayesian networks for complex activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10): 2468–2483, 2013.

[289] Y. Zhang, L. Wang, H. Chen, A. Tian, S. Zhou, and Y. Guo. IF-ConvTransformer: A framework for human activity recognition using IMU fusion and ConvTransformer. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6 (2):1–26, 2022.

[290] Z. Zhang and A. Sarcevic. Constructing awareness through speech, gesture, gaze and movement during a time-critical medical task. In *European Conference on Computer Supported Cooperative Work*, pages 163–182. Springer, 2015.

[291] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang. Deep residual bidir-LSTM for human activity recognition using wearable sensors. *Hindawi Mathematical Problems in Engineering*, 2018(7316954), 2018.

[292] Y. Zhou, Z. Cheng, and L. Jing. Threshold selection and adjustment for online segmentation of one-stroke finger gestures using single tri-axial accelerometer. *Multimedia Tools and Applications*, 74(21):9387–9406, 2015.

[293] C. Zhu and W. Sheng. Motion-and location-based online human daily activity recognition. *Pervasive and Mobile Computing Journal*, 7(2):256–269, 2011.

APPENDICES

Appendix A: Supervisory Evaluation Supplementary Results

The supervisory evaluation results that were not featured in the main chapter are detailed in Appendix A. This supplementary section encompasses the confusion matrices for each component's individual algorithm, along with those for the GNN fusion and TCN concurrent and composite task recognition algorithms.

A.1 Auditory Task Recognition

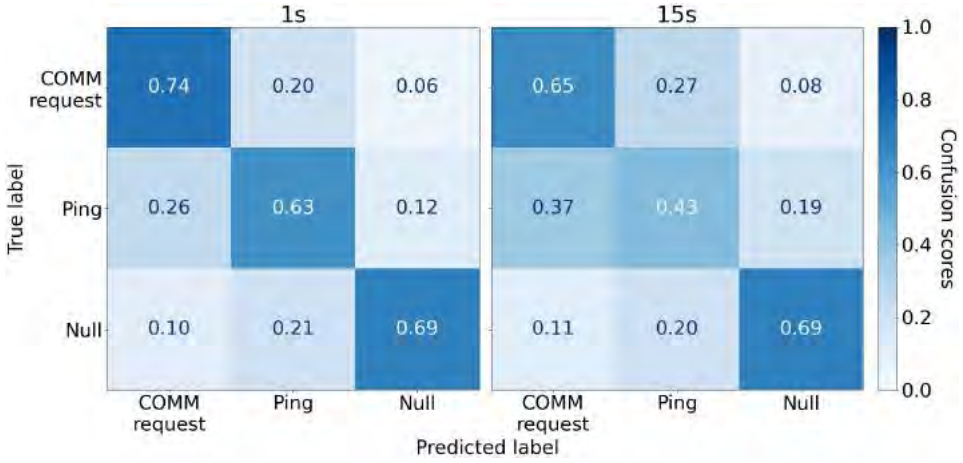


Figure A.1: The RF auditory task recognition algorithm's confusion matrices for the 1s and 15s window sizes.

A.2 Visual Task Recognition

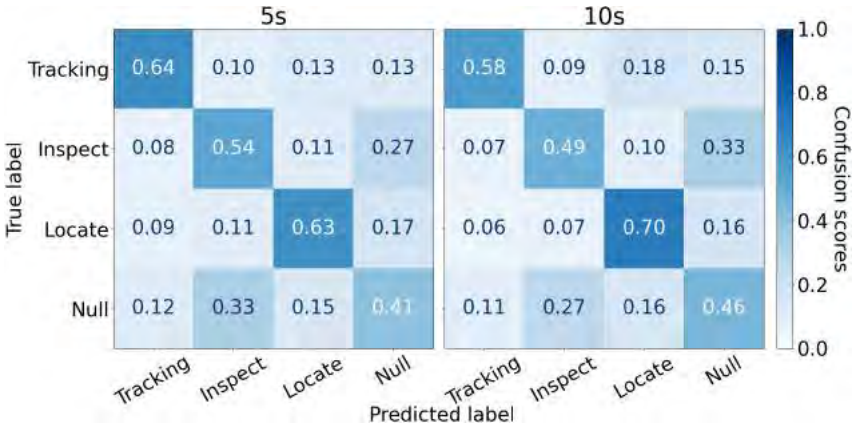


Figure A.2: The visual task recognition confusion matrices when fixation, saccades, and inertial metrics are incorporated for 5s and 10s window sizes.

A.3 Gross Motor Task Recognition

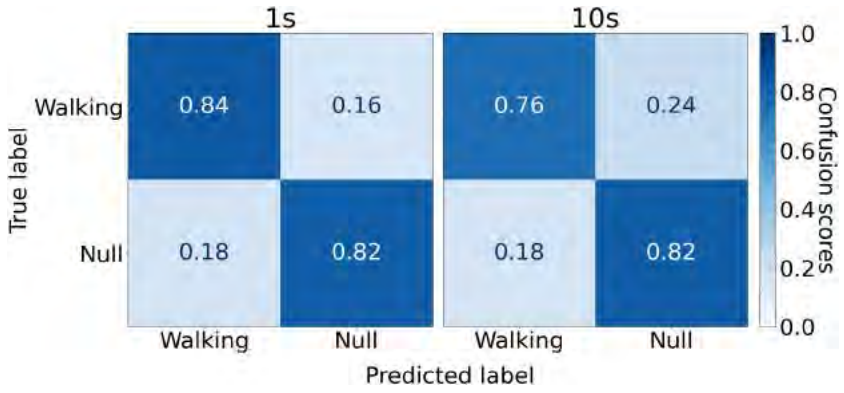


Figure A.3: Gross motor task recognition confusion matrices when incorporating the physiological and four lower-body IMU metrics on both legs for the 1s and 10s window sizes.

Table A.1: Gross motor task recognition accuracy (mean % (std. dev.)) by window size, and incorporated metrics aggregated across participants for the supervisory evaluation. NOTE: The highest accuracy and the corresponding sensor combination are highlighted in Bold. The accuracy when incorporating all metrics is highlighted in Blue, while the overall highest accuracy is in Red.

No. of sensors	**Combination**	**Window size**				
		1s	**2s**	**3s**	**5s**	**10s**
1	Phy	51.39 (3.45)	61.61 (6.41)	61.80 (8.13)	59.70 (9.35)	51.18 (4.32)
	P	77.53 (7.54)	79.40 (7.81)	79.65 (8.20)	79.25 (7.74)	76.82 (8.25)
	T	78.61 (7.05)	79.69 (7.77)	80.33 (7.99)	80.16 (7.84)	77.47 (8.89)
	C	78.17 (7.17)	80.12 (7.48)	80.56 (7.86)	80.15 (7.80)	77.63 (8.10)
	F	78.49 (7.22)	80.18 (7.70)	**80.80 (7.81)**	80.31 (7.57)	76.86 (8.31)
2	Phy + P	77.64 (7.61)	79.54 (7.74)	79.98 (7.68)	79.68 (7.56)	77.16 (8.34)
	Phy + T	78.67 (7.05)	80.49 (7.70)	80.66 (8.34)	80.12 (7.48)	77.51 (9.28)
	Phy + C	78.53 (6.91)	80.42 (7.70)	81.13 (7.86)	80.00 (7.90)	77.92 (7.89)
	Phy + F	78.79 (7.19)	80.54 (7.70)	80.65 (8.33)	80.42 (7.46)	78.12 (8.15)
	P + T	78.69 (7.21)	79.97 (7.62)	80.53 (7.98)	80.16 (7.86)	77.66 (7.71)
	P + C	78.70 (7.24)	80.33 (7.77)	80.91 (7.83)	80.53 (7.56)	77.44 (7.94)
	P + F	79.15 (7.36)	80.76 (7.89)	**81.22 (7.80)**	80.44 (7.76)	77.94 (8.48)
	T + C	78.60 (7.25)	80.55 (7.79)	80.92 (8.10)	80.19 (7.50)	78.24 (8.05)
	T + F	79.32 (7.13)	80.70 (7.89)	81.19 (7.66)	80.51 (7.55)	78.00 (9.48)
	C + F	78.61 (7.27)	80.27 (7.68)	81.18 (7.62)	80.47 (7.53)	77.42 (9.43)
3	Phy + P + T	78.67 (7.24)	80.26 (7.97)	80.67 (8.10)	80.38 (7.61)	77.53 (8.54)
	Phy + P + C	78.75 (7.23)	80.71 (7.98)	80.99 (7.94)	80.53 (7.51)	77.84 (8.25)
	Phy + P + F	78.88 (7.52)	80.91 (8.00)	80.96 (7.91)	80.30 (7.79)	77.70 (8.05)
	Phy + T + C	78.79 (7.08)	80.72 (7.89)	81.02 (8.05)	80.23 (7.93)	78.13 (7.79)
	Phy + T + F	79.27 (7.25)	80.79 (8.07)	80.95 (7.82)	80.42 (7.65)	77.75 (8.02)
	Phy + C + F	78.90 (7.20)	80.74 (7.66)	80.95 (7.89)	80.36 (7.72)	77.72 (7.74)
	P + T + C	78.76 (7.25)	80.74 (7.62)	81.26 (7.76)	80.38 (7.40)	77.55 (8.87)
	P + T + F	79.42 (7.16)	81.05 (7.65)	81.00 (7.41)	80.33 (7.34)	76.82 (8.36)
	P + C + F	78.98 (7.17)	80.68 (7.67)	**81.33 (7.72)**	80.45 (7.26)	77.22 (9.08)
	T + C + F	79.05 (7.21)	81.00 (7.59)	81.16 (7.58)	80.11 (7.46)	77.13 (8.78)
4	Phy + P + T + C	78.81 (7.46)	80.57 (8.10)	81.21 (8.18)	80.23 (7.42)	77.49 (8.35)
	Phy + P + T + F	79.16 (7.52)	80.70 (7.94)	81.09 (7.91)	80.58 (7.46)	78.25 (8.62)
	Phy + P + C + F	78.86 (7.45)	80.91 (7.77)	80.99 (7.99)	80.36 (7.78)	77.75 (8.01)
	Phy + T + C + F	79.27 (7.19)	80.85 (7.62)	81.01 (7.91)	80.23 (7.73)	78.07 (8.48)
	P + T + C + F	79.28 (7.28)	80.98 (7.47)	**81.39 (7.54)**	79.90 (7.36)	75.95 (8.47)
5	**Phy + P + T + C + F**	79.15 (7.55)	80.90 (7.77)	**80.97 (7.89)**	80.19 (7.60)	77.49 (9.53)

A.4 Fine-Grained Motor Task Recognition

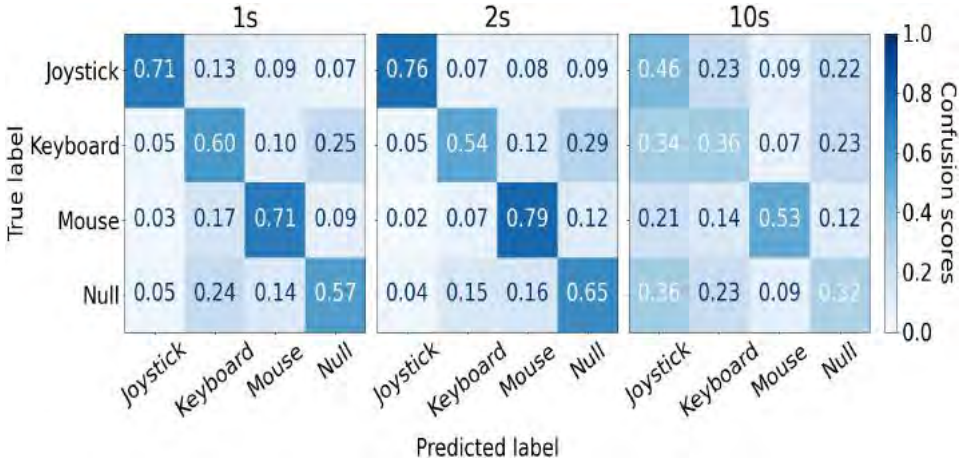


Figure A.4: Fine-grained motor task recognition 1s, 2s and 10s window size confusion matrices when incorporating all four metrics from *both* arms.

Table A.2: Fine-grained motor task recognition accuracy (mean % (std. dev.)) by window size, handedness, and incorporated metrics aggregated across participants for the supervisory evaluation. NOTE: The highest accuracy and the corresponding sensor combination are highlighted in Bold. The accuracy when incorporating all metrics is highlighted in Blue, while the overall highest accuracy is in Red.

No. of sensors	Handedness	Combination	Window size				
			1s	2s	3s	5s	10s
1	Both	F_{imu}	51.14 (16.04)	52.19 (17.47)	54.59 (18.51)	49.18 (17.27)	47.05 (15.54)
		H	61.26 (8.54)	65.24 (9.57)	**68.56 (9.26)**	68.24 (8.90)	65.60 (9.55)
		W	60.53 (9.41)	64.44 (10.62)	66.60 (10.66)	66.69 (9.46)	64.55 (10.52)
		F_{emg}	49.17 (15.18)	50.51 (16.38)	50.04 (16.49)	46.61 (15.06)	40.50 (11.19)
	Left	F_{imu}	38.68 (9.35)	39.90 (10.80)	40.34 (11.00)	42.75 (12.86)	43.52 (12.57)
		H	48.11 (6.83)	51.64 (8.09)	53.08 (8.38)	55.39 (8.88)	53.46 (8.06)
		W	50.18 (7.13)	52.64 (7.88)	54.97 (8.35)	56.56 (8.39)	57.51 (8.76)
		F_{emg}	33.78 (9.34)	34.56 (9.74)	34.21 (9.14)	33.77 (9.92)	31.96 (8.41)
	Right	F_{imu}	48.48 (15.97)	49.76 (16.93)	49.85 (16.92)	48.19 (16.50)	44.23 (12.75)
		H	56.19 (7.66)	59.39 (9.74)	61.53 (9.77)	62.98 (9.50)	58.69 (10.48)
		W	57.99 (10.00)	60.77 (10.28)	62.85 (10.86)	62.48 (10.79)	59.81 (12.67)
		F_{emg}	45.16 (13.90)	47.10 (15.41)	46.94 (15.70)	45.40 (15.81)	37.67 (10.62)
2	Both	F_{imu} + H	59.21 (11.37)	62.62 (10.84)	65.37 (10.63)	64.91 (10.03)	60.37 (10.31)
		F_{imu} + W	59.36 (11.18)	64.33 (11.85)	65.05 (12.08)	64.51 (11.68)	59.39 (9.84)
		F_{imu} + F_{emg}	55.76 (19.65)	56.67 (20.37)	56.23 (20.33)	51.05 (18.36)	47.21 (16.09)
		H + F_{emg}	65.16 (11.50)	68.11 (12.19)	**69.29 (11.84)**	68.34 (11.55)	65.66 (9.23)
		W + H	62.23 (9.31)	66.18 (10.58)	68.51 (10.94)	68.07 (9.08)	64.80 (8.86)
		W + F_{emg}	64.50 (12.75)	67.75 (13.01)	68.47 (13.41)	66.90 (11.38)	62.82 (10.74)
	Left	F_{imu} + H	43.13 (8.12)	47.32 (8.00)	49.28 (9.12)	52.26 (9.79)	51.15 (8.64)
		F_{imu} + W	47.07 (7.56)	50.67 (8.16)	51.92 (8.23)	54.17 (9.04)	50.94 (8.36)
		F_{imu} + F_{emg}	40.42 (10.67)	40.12 (10.76)	40.85 (11.40)	42.42 (13.25)	41.53 (13.06)
		H + F_{emg}	48.52 (8.16)	51.25 (8.78)	52.55 (9.20)	54.24 (8.28)	50.76 (8.81)
		W + H	51.16 (7.32)	54.47 (7.72)	56.37 (8.55)	57.50 (8.44)	54.42 (8.69)
		W + F_{emg}	50.55 (7.50)	53.22 (8.29)	55.05 (9.20)	56.19 (9.26)	52.97 (7.34)
	Right	F_{imu} + H	54.34 (10.96)	58.78 (11.75)	58.75 (12.04)	59.48 (10.18)	56.71 (10.45)
		F_{imu} + W	56.38 (10.39)	59.41 (11.35)	59.91 (10.94)	59.86 (11.42)	57.86 (9.88)
		F_{imu} + F_{emg}	51.40 (17.87)	52.69 (19.19)	53.86 (19.44)	49.53 (18.73)	45.88 (16.17)
		H + F_{emg}	58.34 (10.46)	61.96 (10.91)	62.98 (11.98)	63.38 (11.51)	60.90 (12.42)
		W + H	59.67 (8.89)	62.53 (9.28)	63.52 (10.60)	64.04 (9.76)	59.63 (10.92)
		W + F_{emg}	60.15 (11.03)	64.35 (11.12)	65.77 (12.10)	64.77 (12.06)	59.60 (12.48)
3	Both	F_{imu} + H + F_{emg}	61.45 (16.01)	62.59 (14.75)	66.21 (14.25)	65.86 (12.63)	61.25 (11.53)
		F_{imu} + W + H	61.03 (10.59)	65.35 (11.14)	67.31 (11.29)	66.94 (10.07)	61.99 (9.66)
		F_{imu} + W + F_{emg}	61.81 (15.67)	66.04 (15.98)	66.70 (15.01)	64.83 (13.70)	58.84 (12.02)
		W + H + F_{emg}	66.68 (11.80)	69.52 (12.15)	**70.04 (13.04)**	69.25 (10.54)	63.75 (9.75)
	Left	F_{imu} + H + F_{emg}	45.58 (8.76)	46.84 (8.21)	49.47 (9.51)	51.83 (8.56)	49.63 (7.96)
		F_{imu} + W + H	48.31 (8.05)	51.39 (8.48)	53.74 (8.64)	54.47 (9.40)	51.21 (8.45)
		F_{imu} + W + F_{emg}	47.06 (8.07)	50.66 (8.50)	52.60 (9.10)	53.19 (8.63)	46.74 (7.82)
		W + H + F_{emg}	50.99 (7.56)	54.80 (7.94)	56.38 (9.45)	56.65 (9.75)	52.60 (9.10)
	Right	F_{imu} + H + F_{emg}	57.51 (13.55)	60.71 (13.48)	61.64 (13.12)	60.36 (13.62)	58.60 (12.36)
		F_{imu} + W + H	58.00 (9.37)	61.31 (10.34)	61.98 (10.73)	62.75 (10.06)	58.90 (9.93)
		F_{imu} + W + F_{emg}	58.17 (14.24)	61.63 (13.51)	63.16 (12.81)	62.33 (12.72)	56.89 (12.29)
		W + H + F_{emg}	61.32 (10.88)	65.06 (11.13)	67.38 (11.16)	66.00 (11.47)	60.00 (12.24)
4	Both	**F_{imu} + W + H + F_{emg}**	64.68 (14.95)	66.89 (14.84)	**68.57 (14.00)**	66.02 (12.08)	62.09 (10.93)
	Left	F_{imu} + W + H + F_{emg}	49.08 (7.73)	51.27 (7.72)	53.90 (9.75)	52.88 (9.29)	50.06 (8.53)
	Right	F_{imu} + W + H + F_{emg}	58.44 (13.73)	63.06 (13.19)	64.62 (12.48)	62.86 (12.10)	58.22 (11.39)

A.5 Tactile Task Recognition

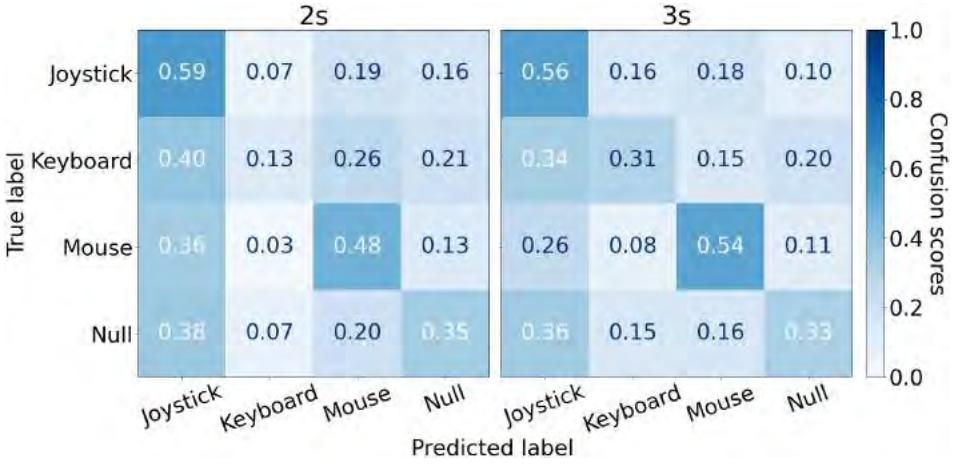


Figure A.5: Tactile task recognition confusion matrices when IMU and sEMG metrics are incorporated on *Both* hands for 2s and 3s window sizes.

Table A.3: Tactile task recognition accuracy (mean % (std. dev.)) by window size, handedness, and incorporated metrics aggregated across participants for the supervisory evaluation. NOTE: The highest accuracy when incorporating a single metric is highlighted in Bold, while the overall highest accuracy is in Blue.

No. of sensors	Handedness	Combination	Window size				
			0.5s	1s	1.5s	2s	3s
1	Both	**H**	62.00 (8.36)	64.37 (7.83)	**65.18 (7.22)**	64.15 (8.08)	60.03 (12.45)
		F_{emg}	51.56 (15.68)	51.69 (17.53)	47.66 (15.54)	41.06 (13.51)	36.78 (16.71)
	Left	H	49.48 (8.97)	52.38 (9.90)	51.47 (10.20)	50.92 (9.70)	46.66 (10.60)
		F_{emg}	34.68 (9.68)	35.96 (10.95)	33.32 (9.90)	30.55 (9.73)	29.17 (10.62)
	Right	H	55.10 (8.69)	58.29 (8.76)	58.88 (9.66)	58.81 (10.92)	57.02 (16.79)
		F_{emg}	46.46 (13.94)	46.53 (16.27)	43.70 (13.86)	38.88 (13.23)	34.14 (11.69)
2	Both	**H + F_{emg}**	66.52 (13.72)	**68.06 (12.93)**	67.45 (11.53)	65.24 (10.78)	61.68 (14.11)
	Left	H + F_{emg}	50.39 (9.84)	52.45 (8.95)	49.39 (10.73)	49.15 (9.70)	43.63 (14.73)
	Right	H + F_{emg}	61.16 (11.86)	61.14 (12.79)	61.26 (12.94)	58.29 (11.33)	53.42 (15.62)

A.6 GNN Fusion Task Recognition

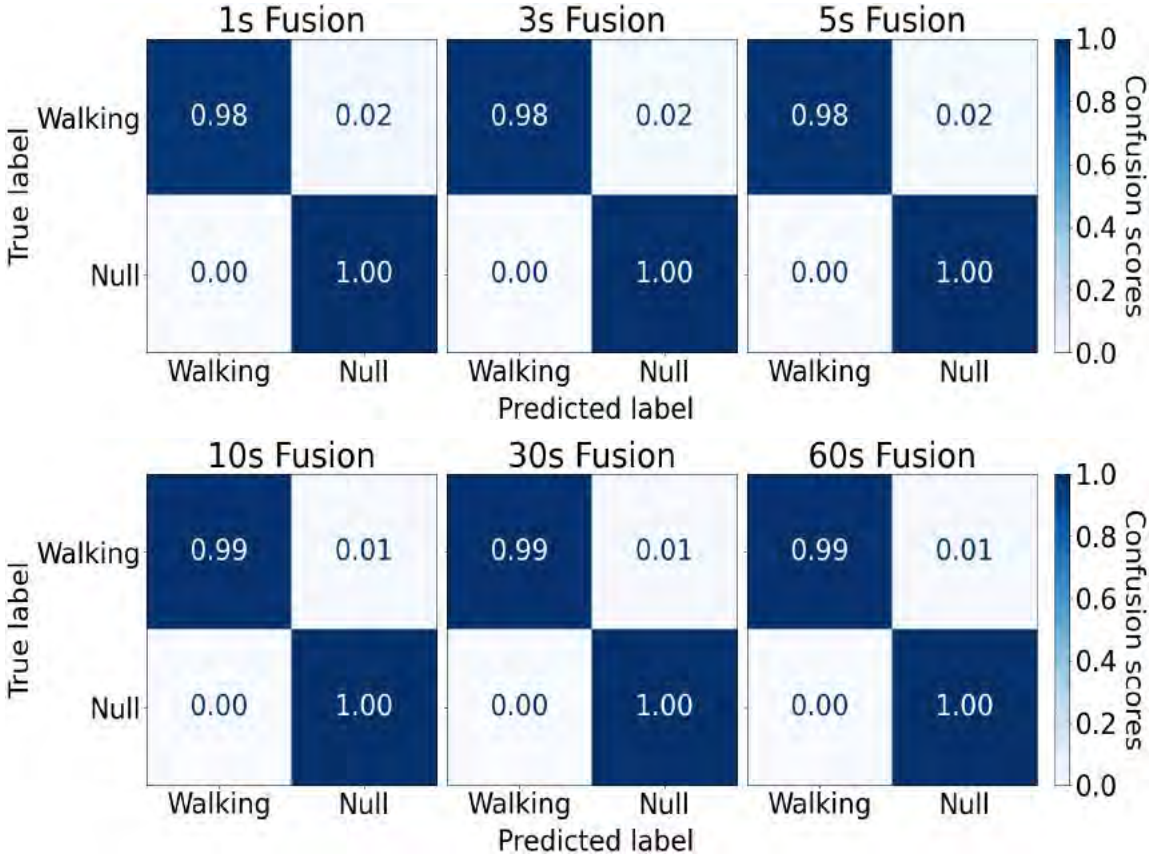


Figure A.6: Gross motor component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

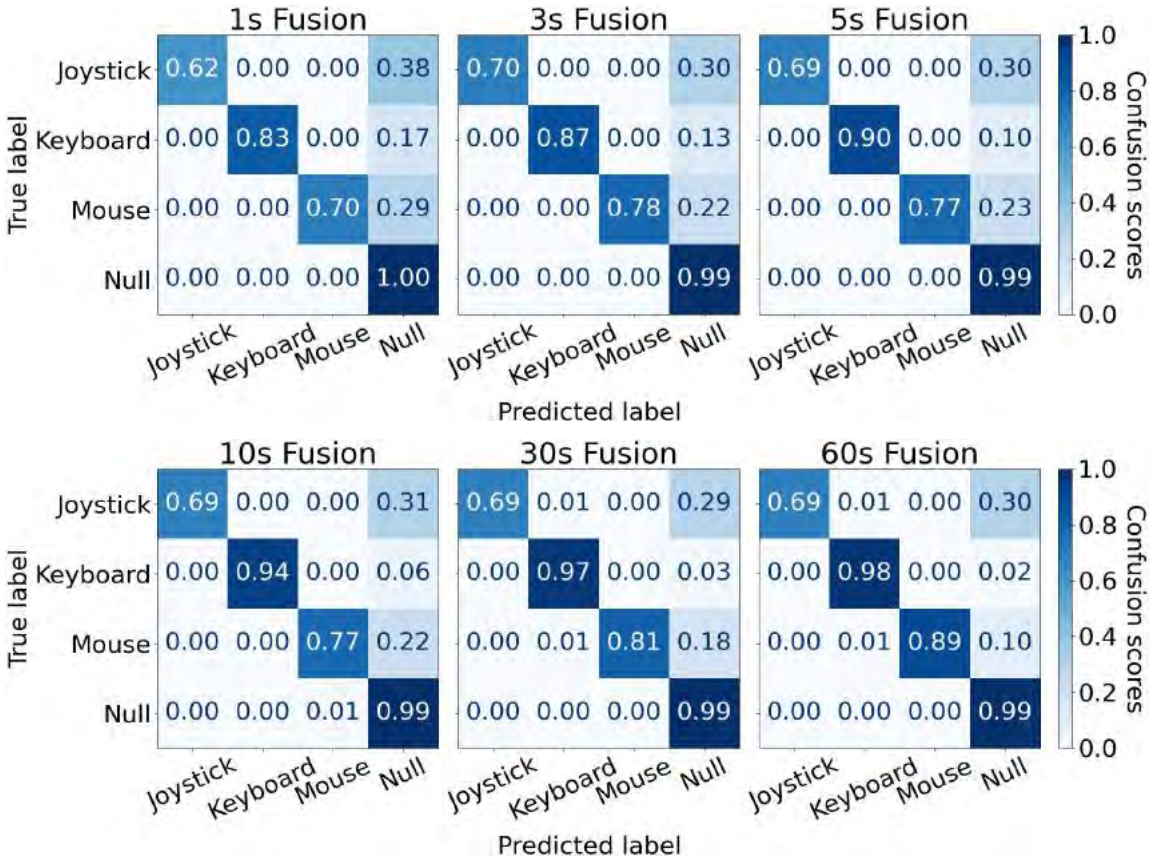


Figure A.7: Fine-grained motor component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

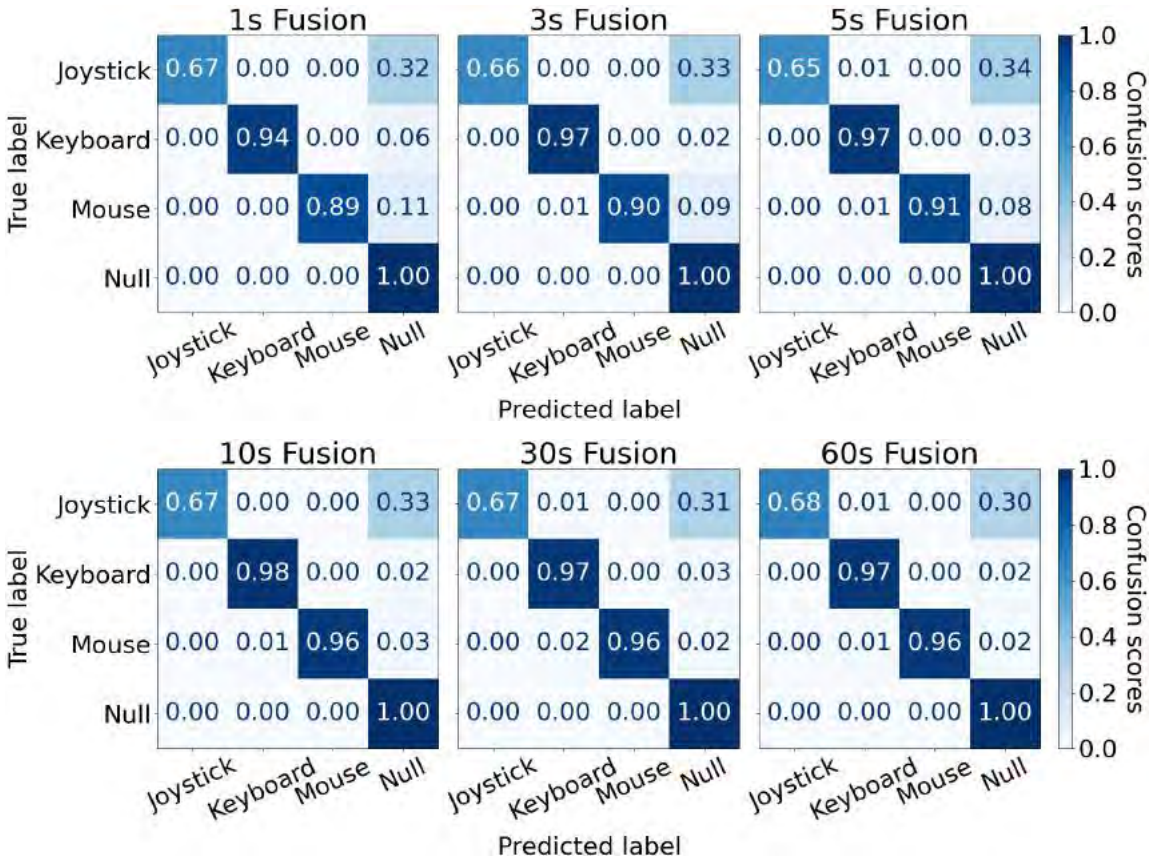


Figure A.8: Tactile component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

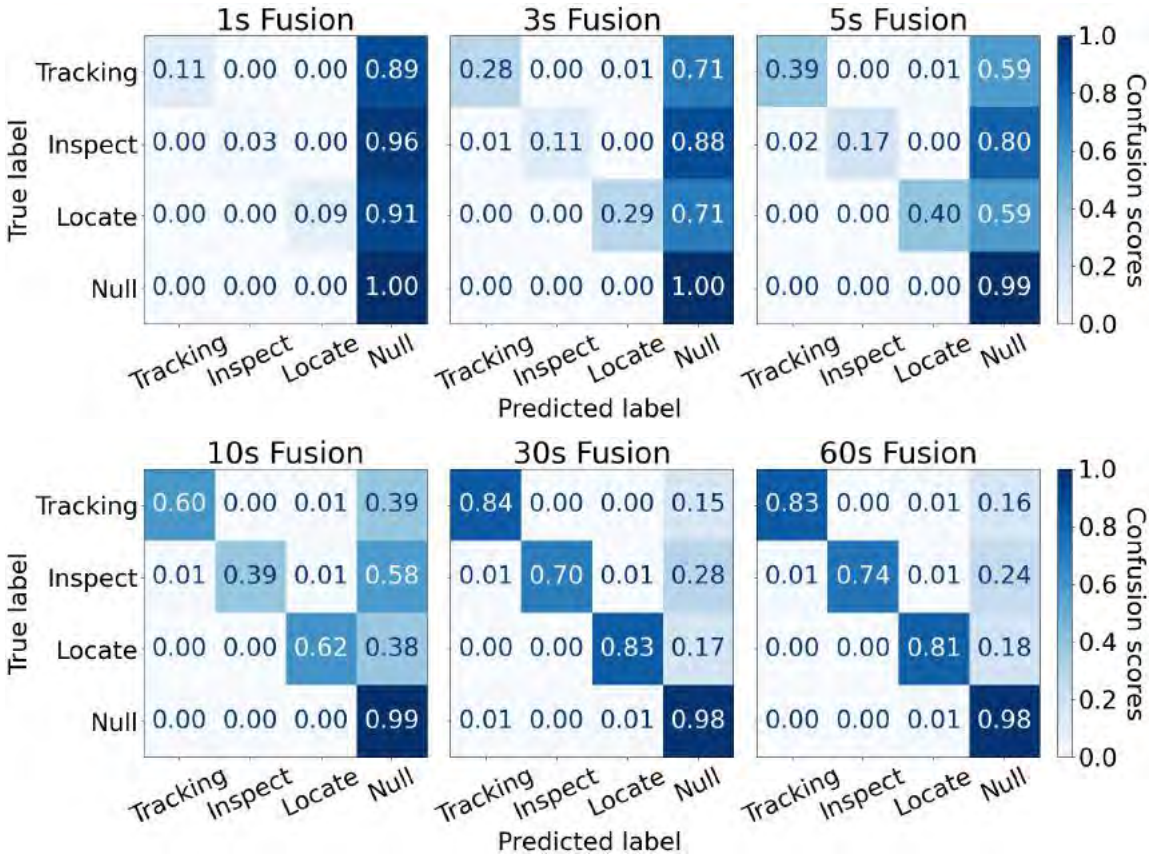


Figure A.9: Visual component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

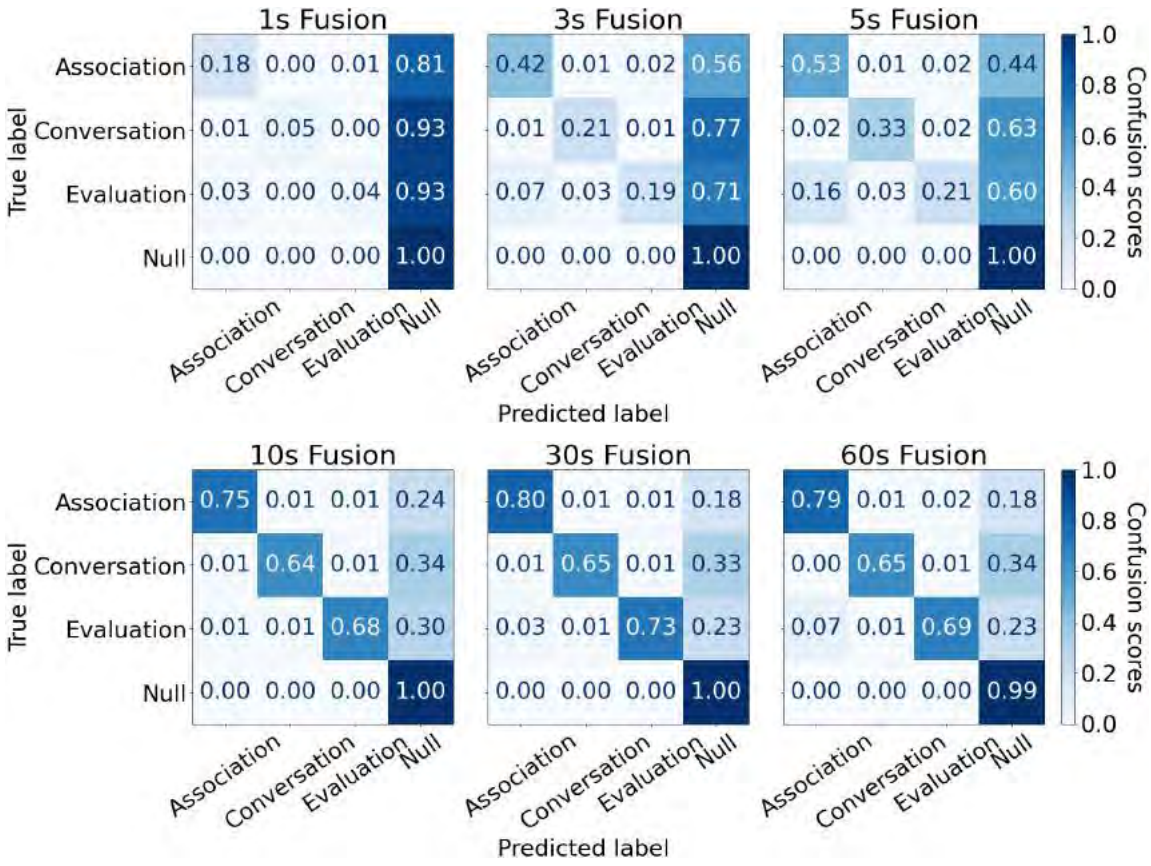


Figure A.10: Cognitive component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

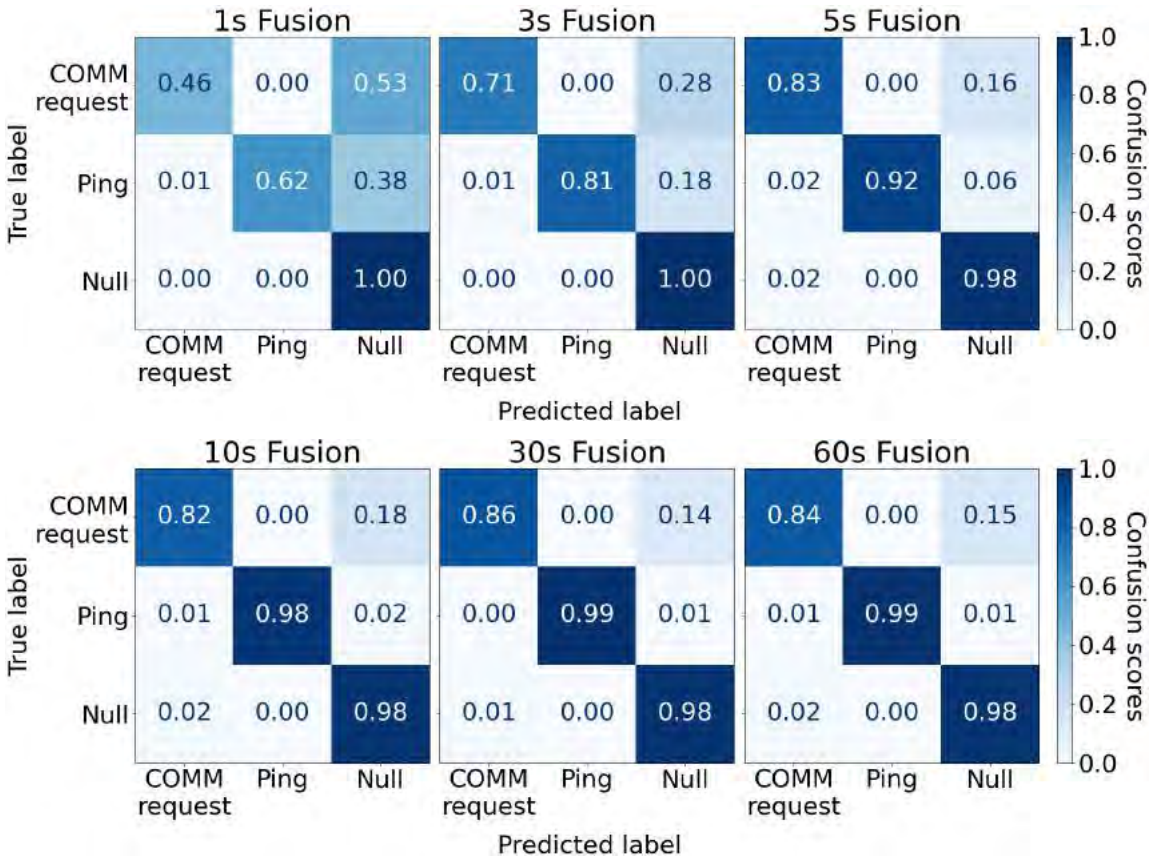


Figure A.11: Auditory component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

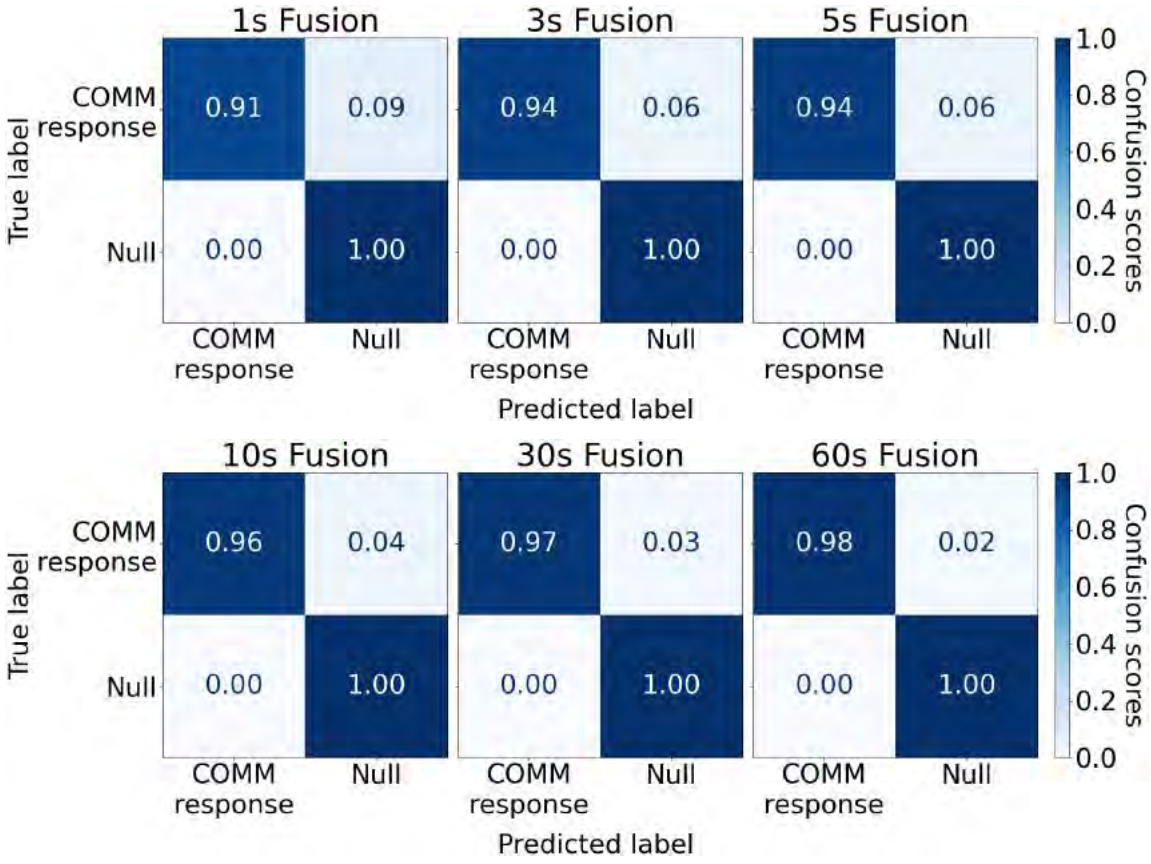
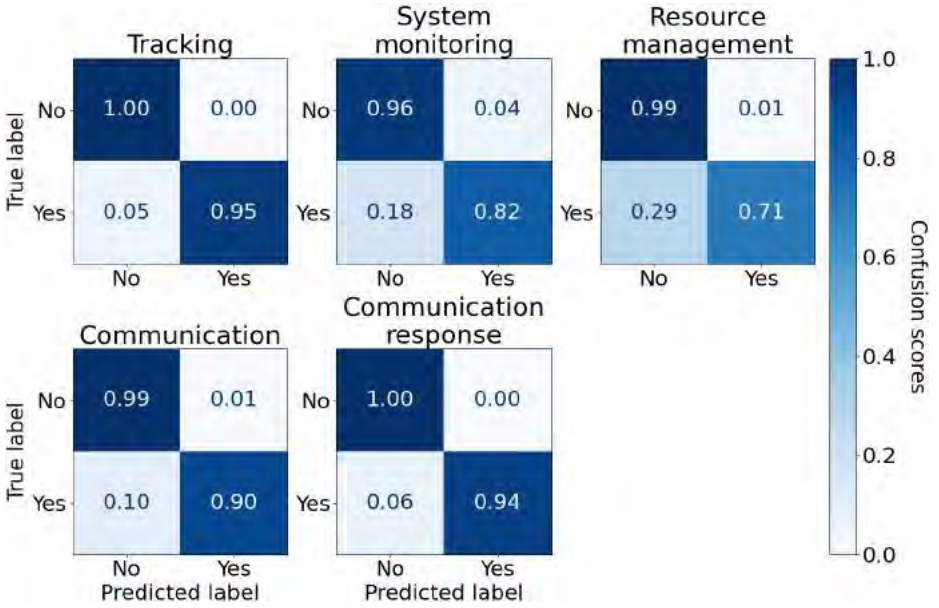
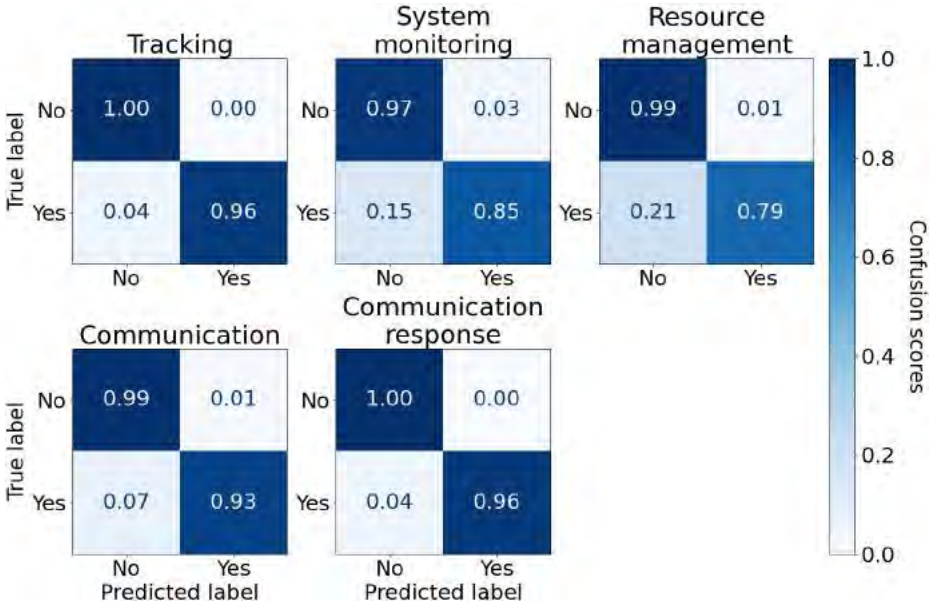


Figure A.12: Speech component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 5s, 10s, 30s, and 60s window sizes.

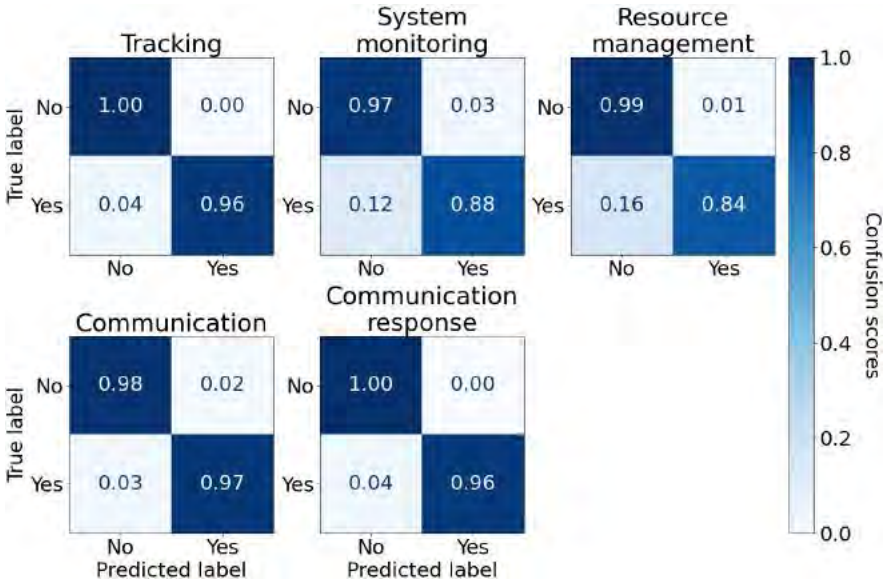
A.7 Composite and Concurrent Task Recognition



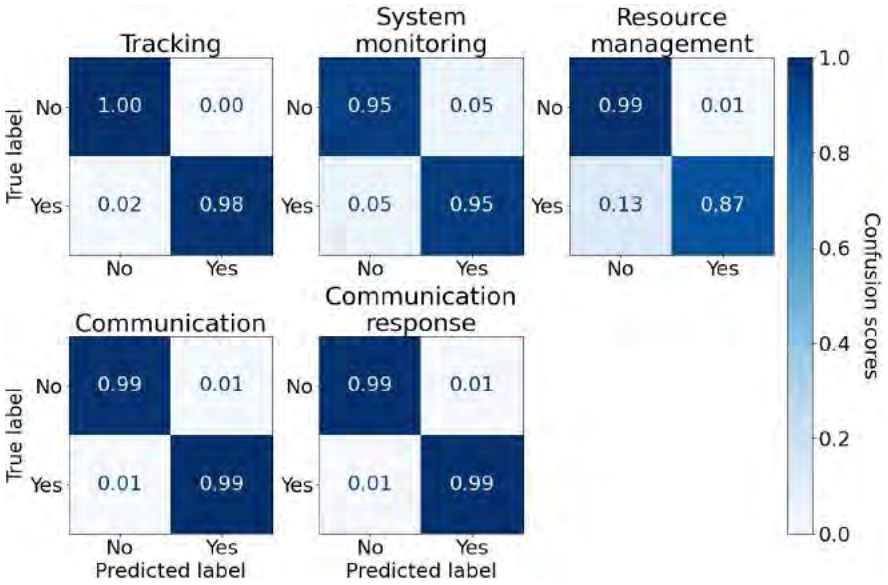
(a) 3s window size



(b) 5s window size



(c) 10s window size



(d) 30s window size

Figure A.13: TCN composite and concurrent task recognition algorithm's composite task' multi-label for the 3s, 5s, 10s, and 30s window sizes.

Appendix B: Peer-Based Evaluation Supplementary Results

Peer-based evaluation results that were not presented in the main chapter are provided in Appendix B. These results encompass the confusion matrices for each component's individual algorithm, along with the confusion matrices for the GNN fusion and TCN concurrent and composite task recognition algorithms.

B.1 Speech Task Recognition

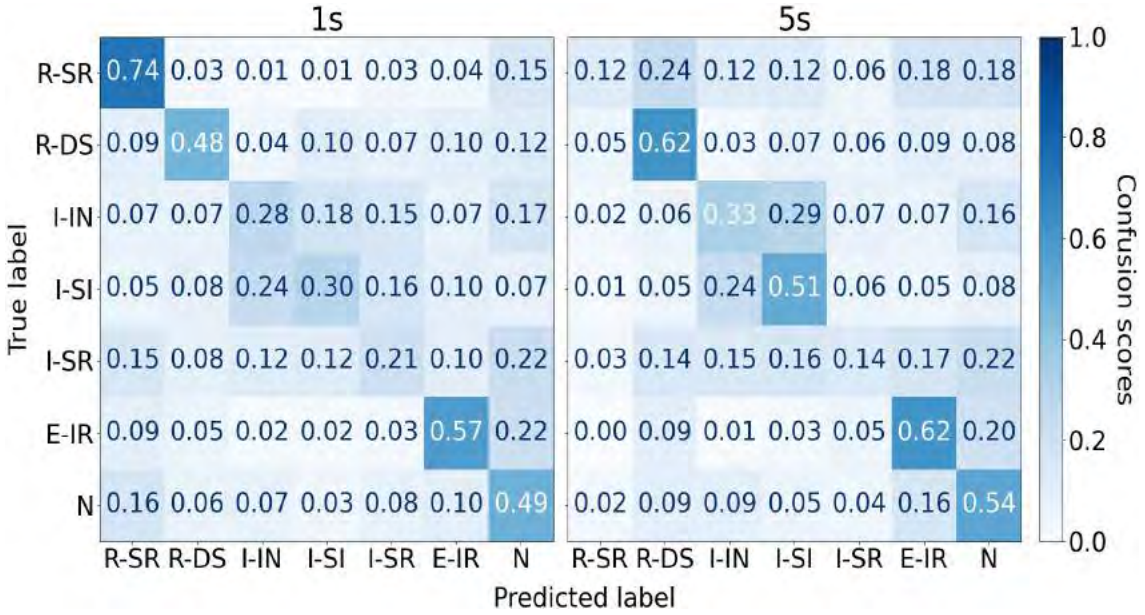


Figure B.1: Speech-reliant task recognition confusion matrix for the 1s and 5s window sizes. Reminder: *R-SR*: Requesting robot to scan an item, *R-DS*: Describing sample to the robot, *I-IN*: Providing information to the Incident Commander, *I-SI*: Describing a suspicious item to the Incident Commander, *I-SR*: Responding to Incident Commander's secondary prompt, *E-IR*: Responding to experimenter's in-situ probe, and *N*: Null.

B.2 Auditory Task Recognition

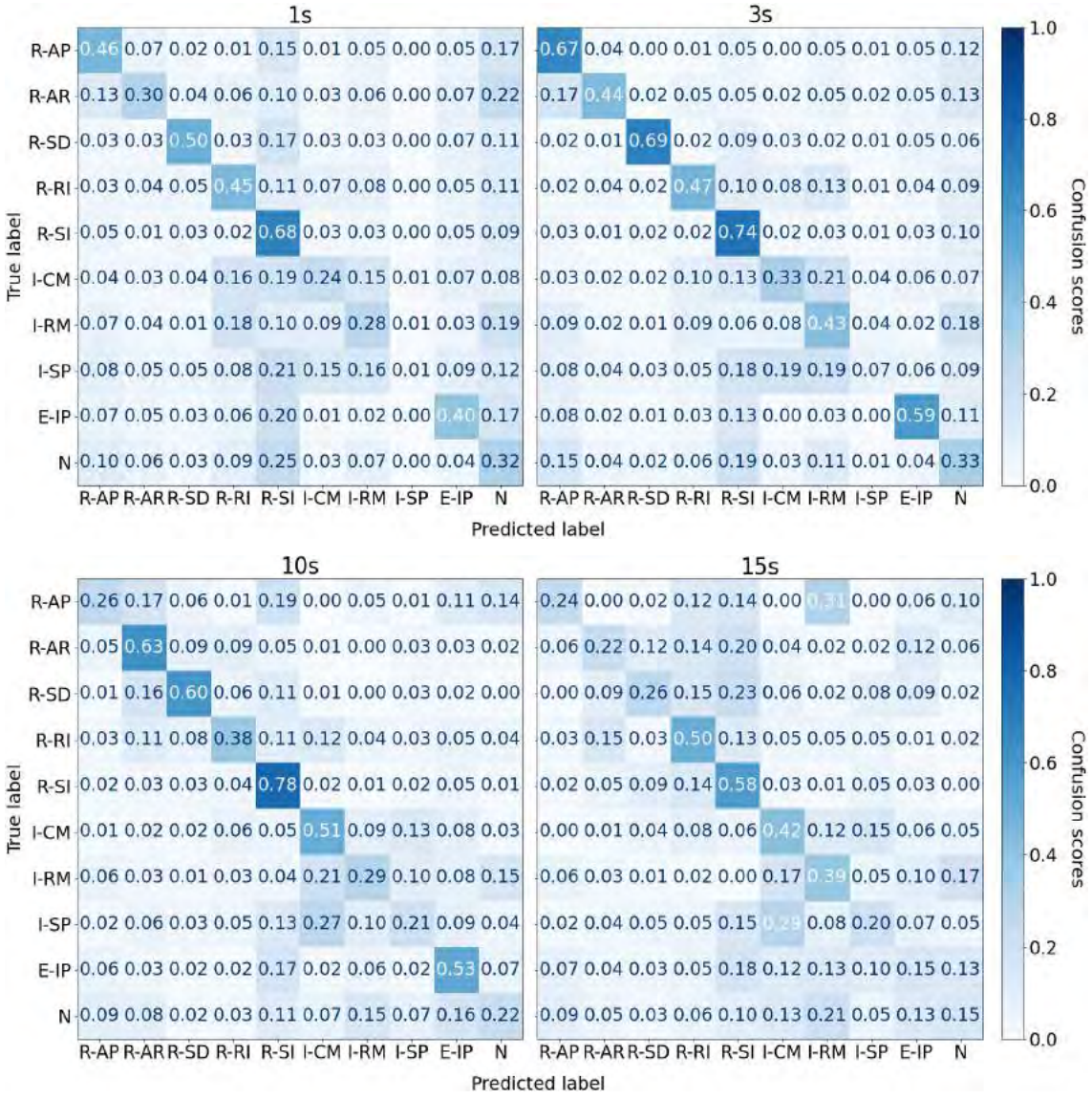


Figure B.2: Auditory task recognition confusion matrix for the 1s, 3s, 10s and 15s window sizes. Reminder: *R-AP*: Robot's analyze prompt, *R-AR*: Robot's assist request, *R-SD*: Robot's sample description request, *R-RI*: Robot's report to Incident Commander prompt, *R-SI*: Robot's sampling instructions, *I-CM*: Incident Commander's communication, *I-RM*: Incident Commander's reminder, *I-SP*: Incident Commander's secondary prompt, *E-IP*: Experimenter's in-situ probe, and *N*: Null.

B.3 Gross Motor Task Recognition

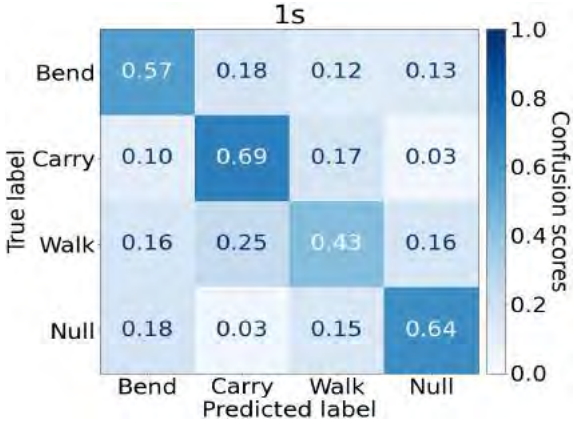


Figure B.3: Gross motor task recognition confusion matrices when incorporating the physiological and IMU metrics for the 1s window size.

B.4 Tactile Task Recognition

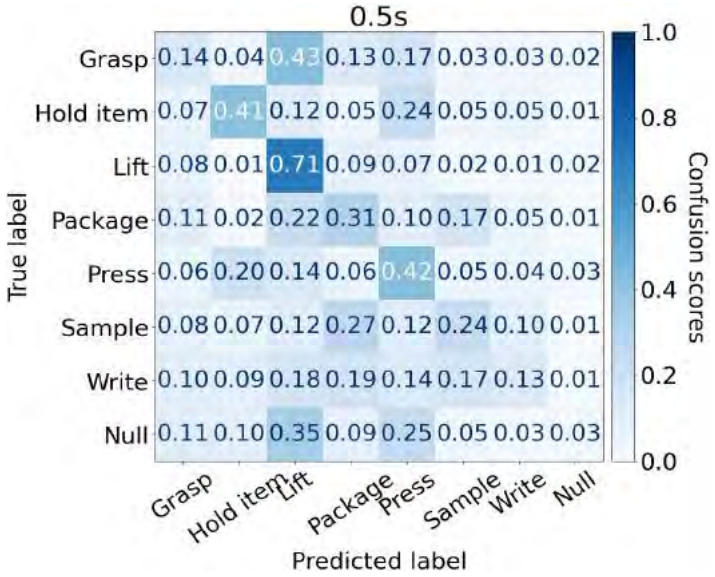


Figure B.4: Tactile task recognition confusion matrices when IMU and sEMG metrics are incorporated on *Both* hands for the 0.5s window size.

B.5 GNN Fusion Task Recognition

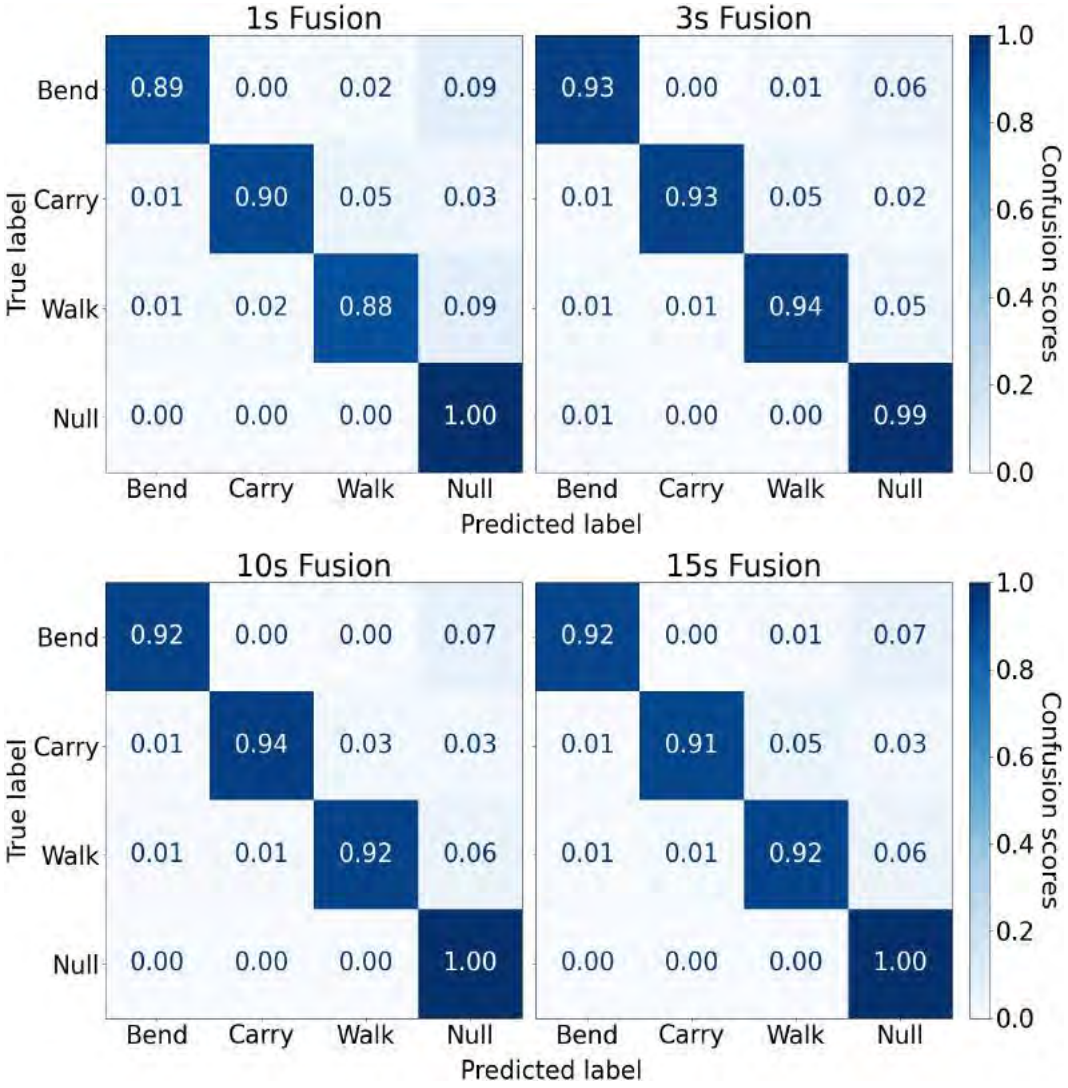


Figure B.5: Gross motor component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

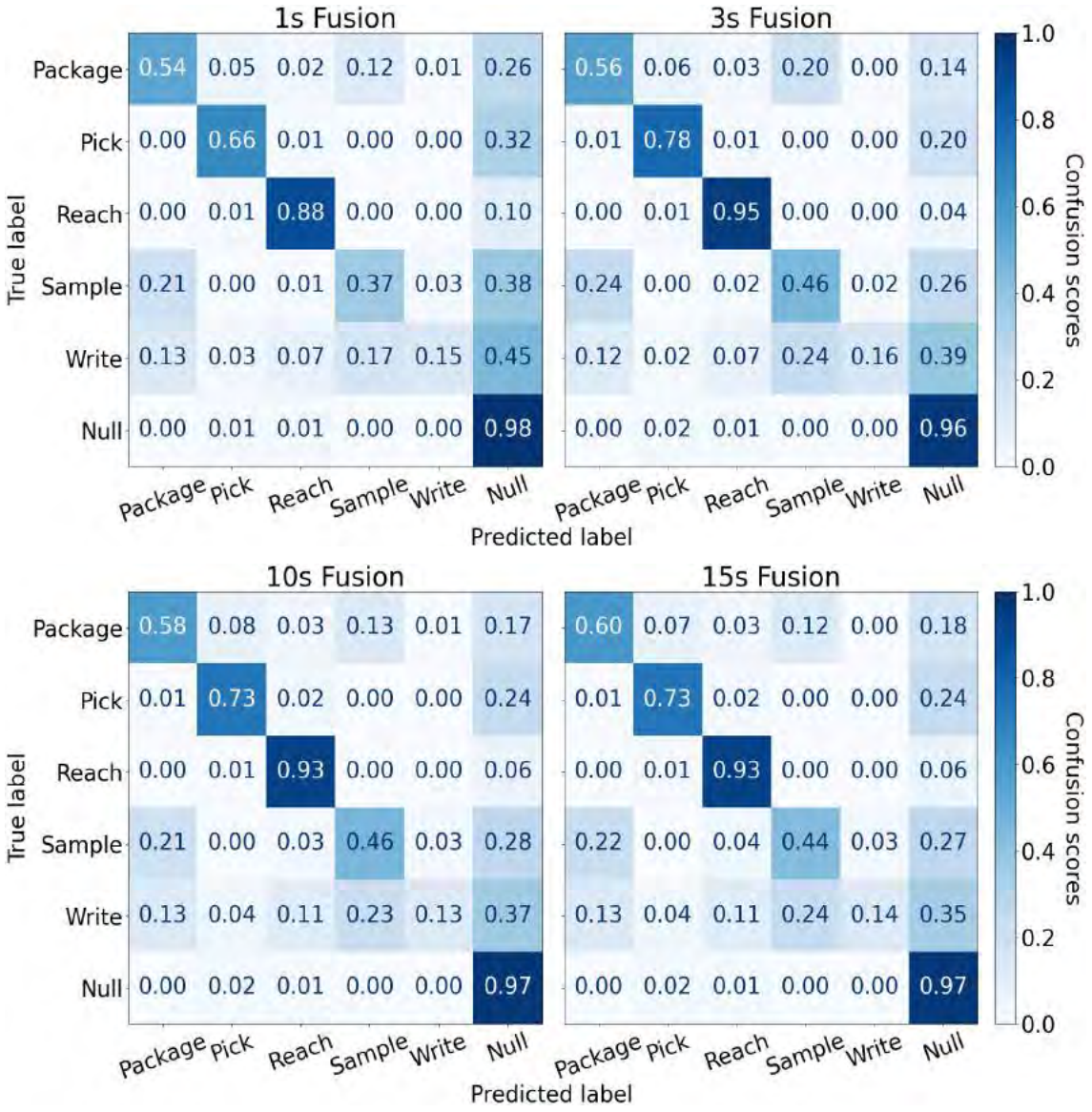


Figure B.6: Fine-grained motor component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

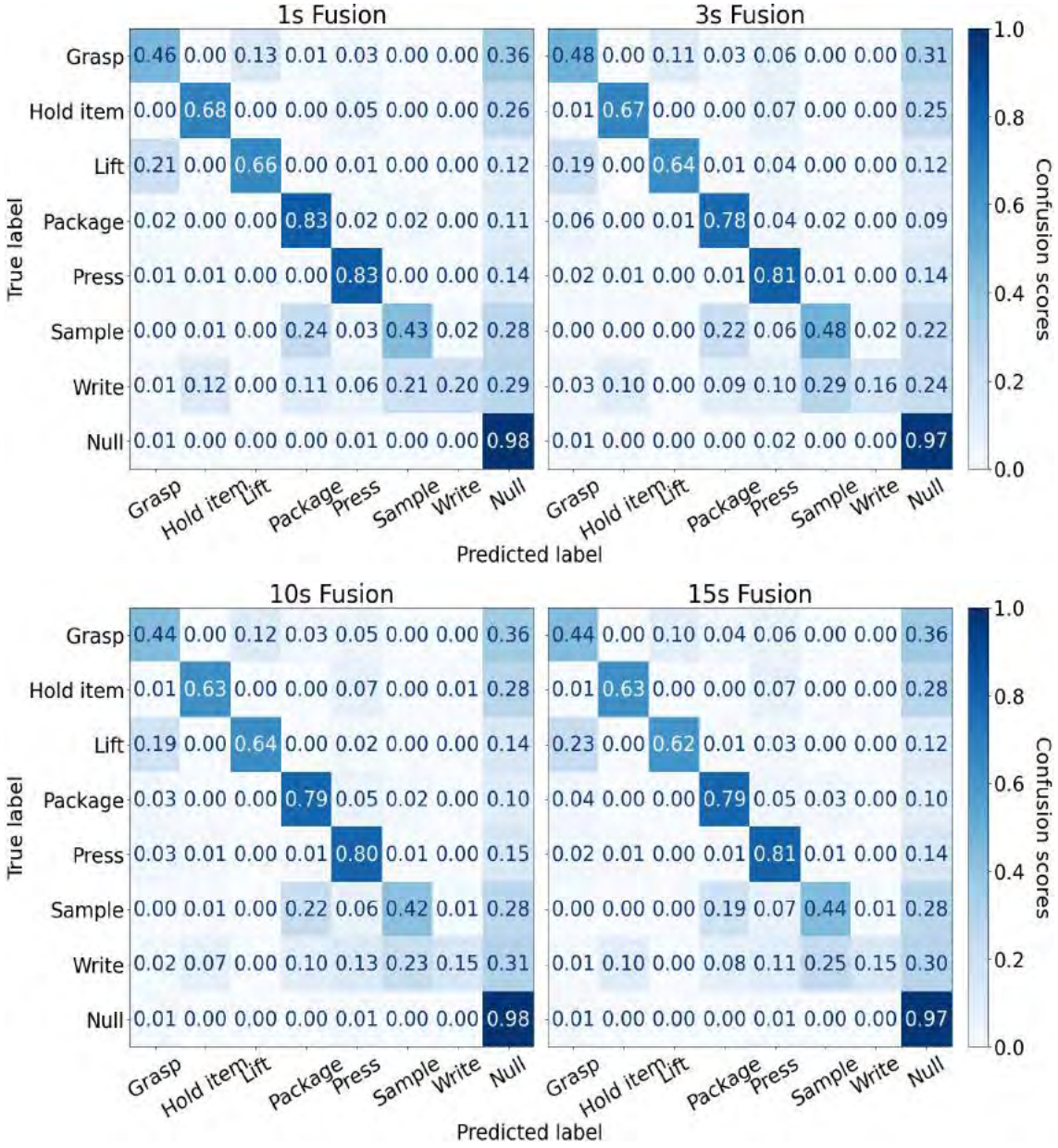


Figure B.7: Tactile component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

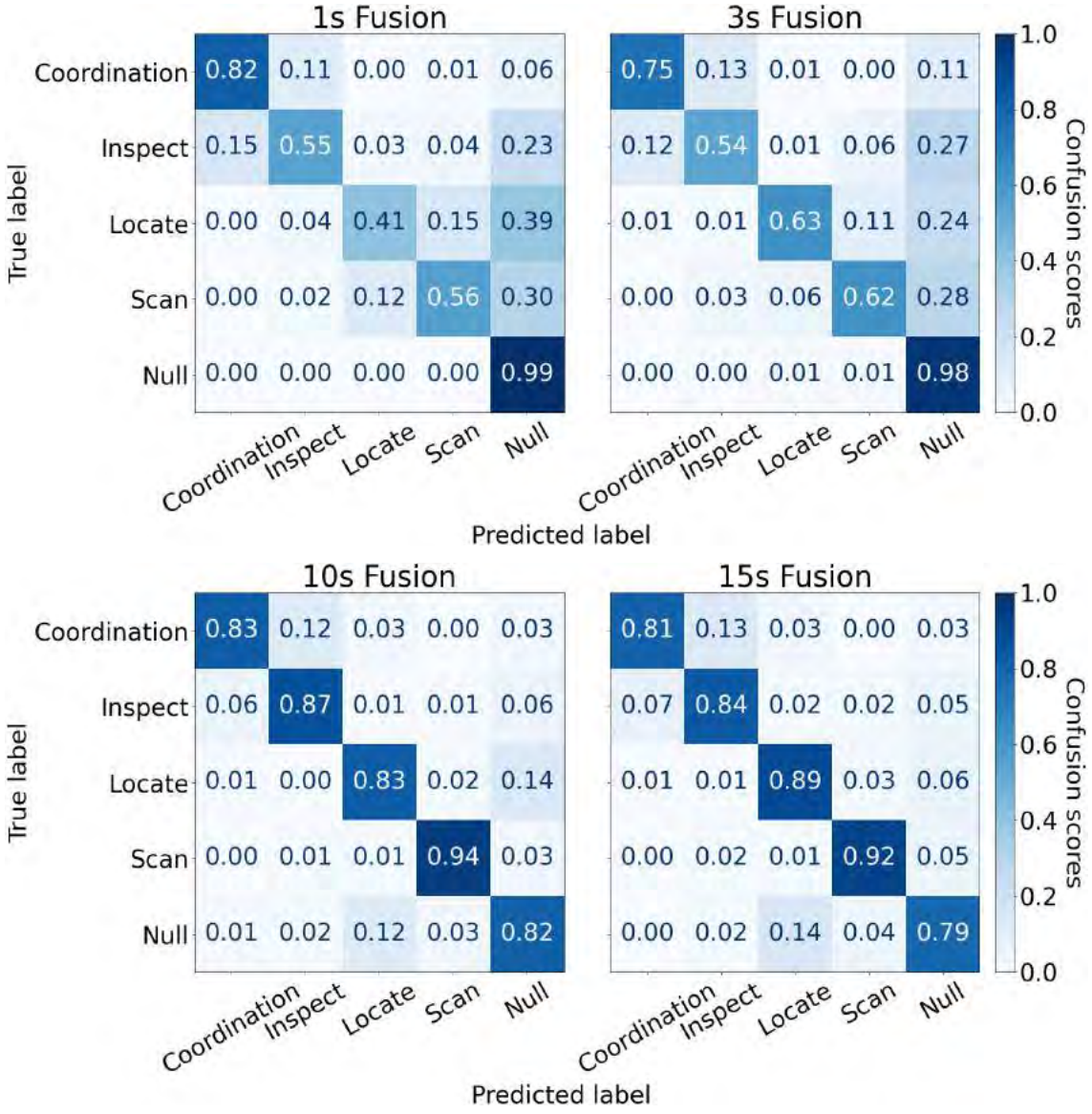


Figure B.8: Visual component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

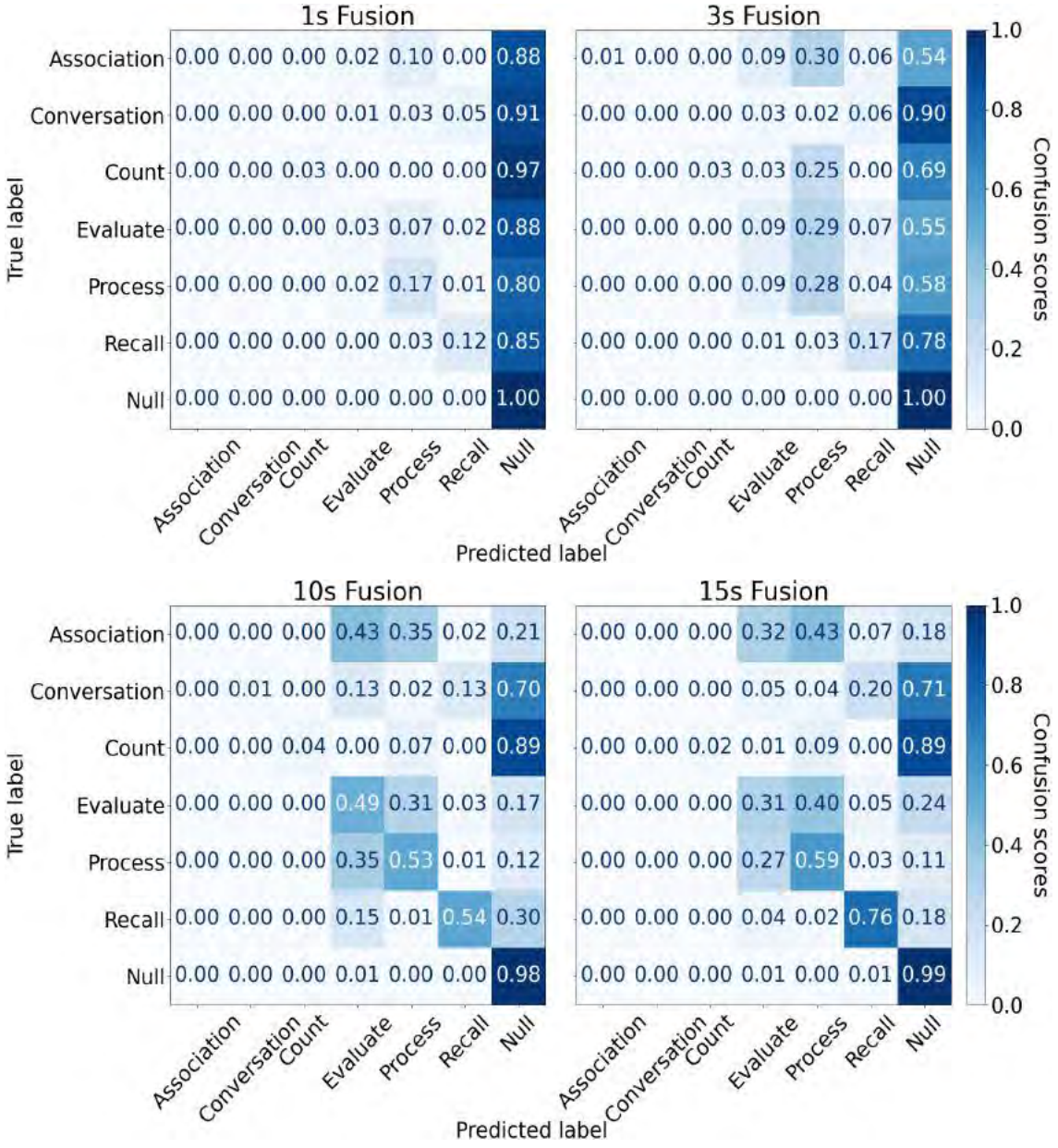


Figure B.9: Cognitive component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

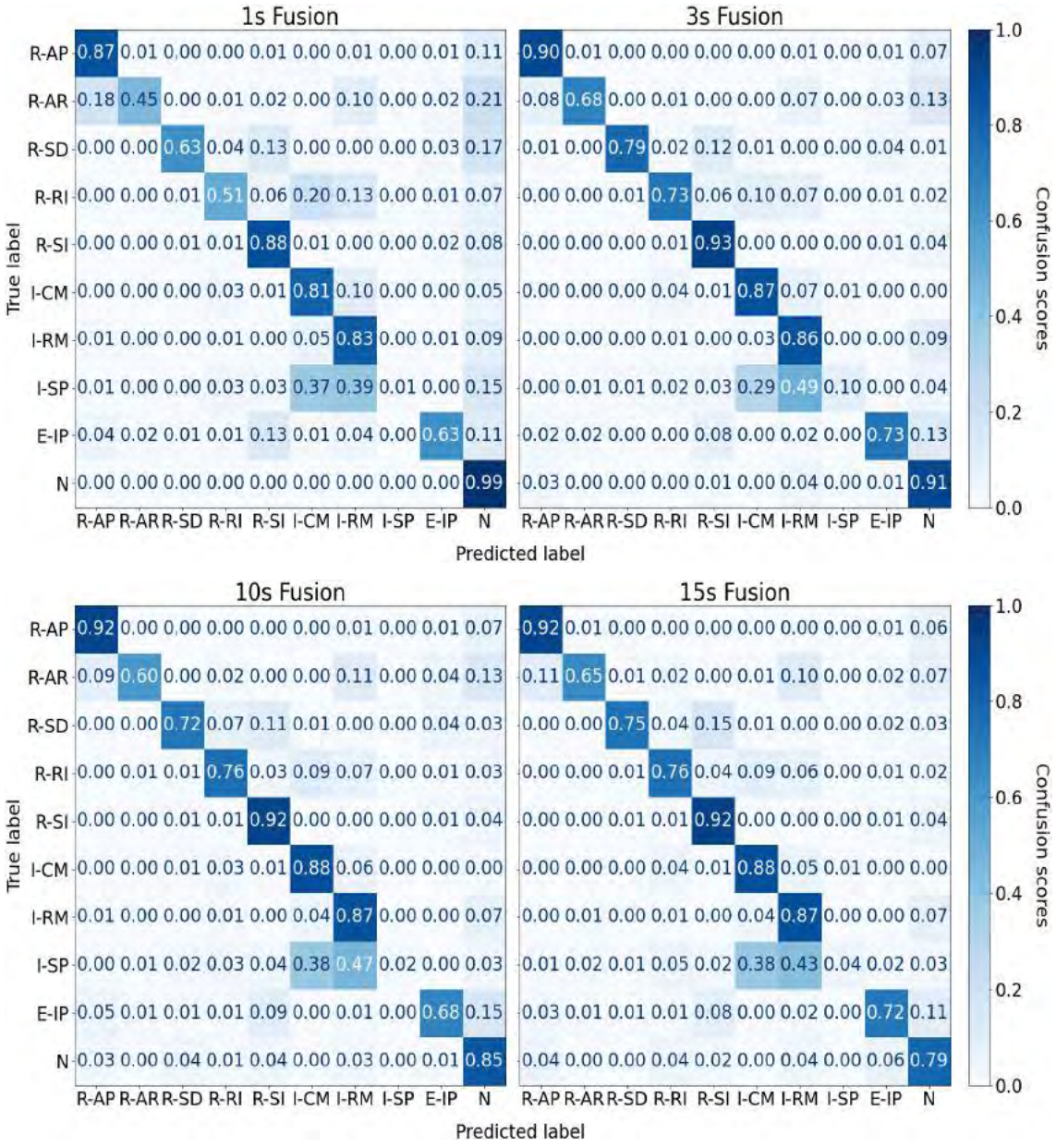


Figure B.10: Auditory component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

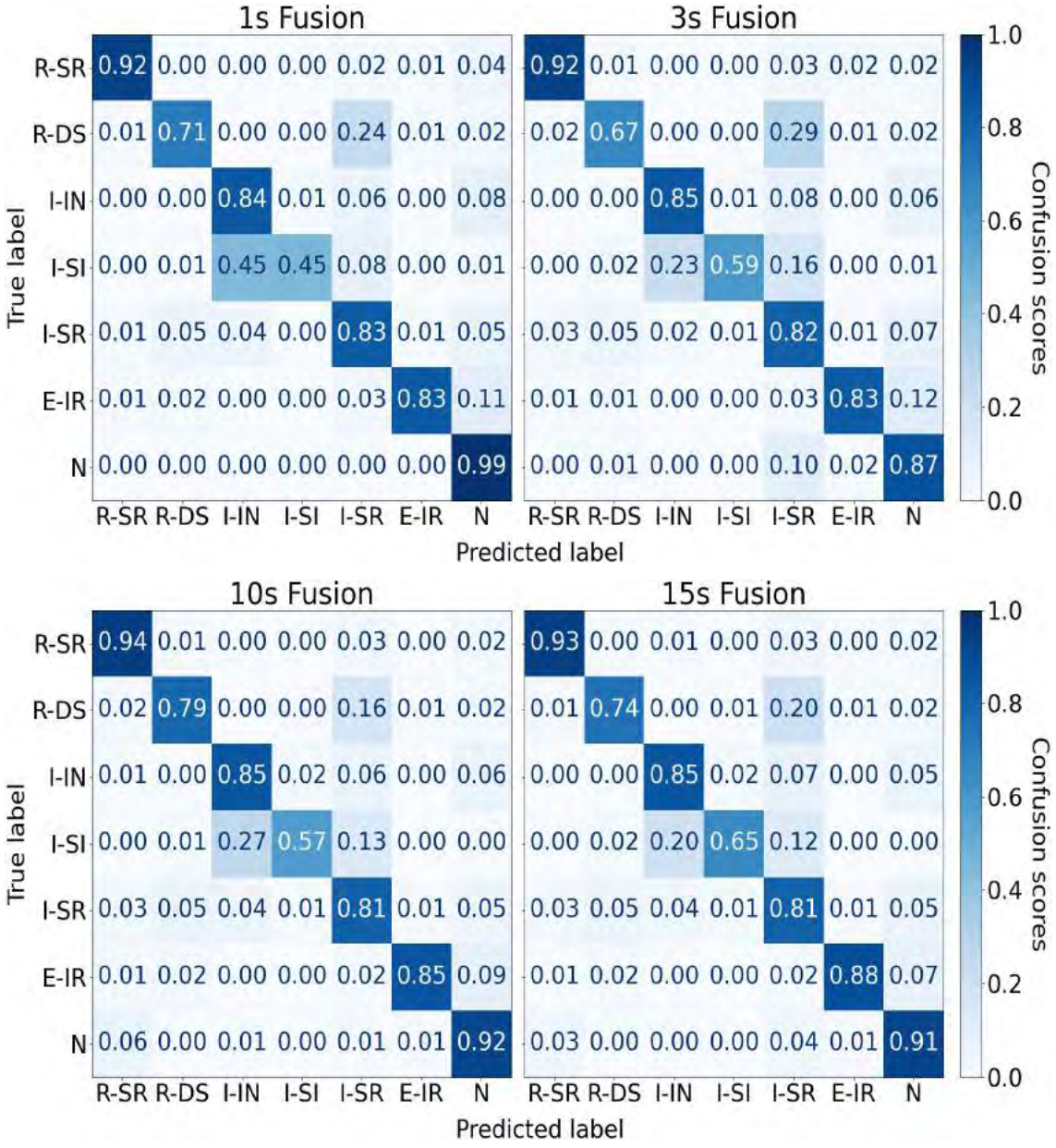
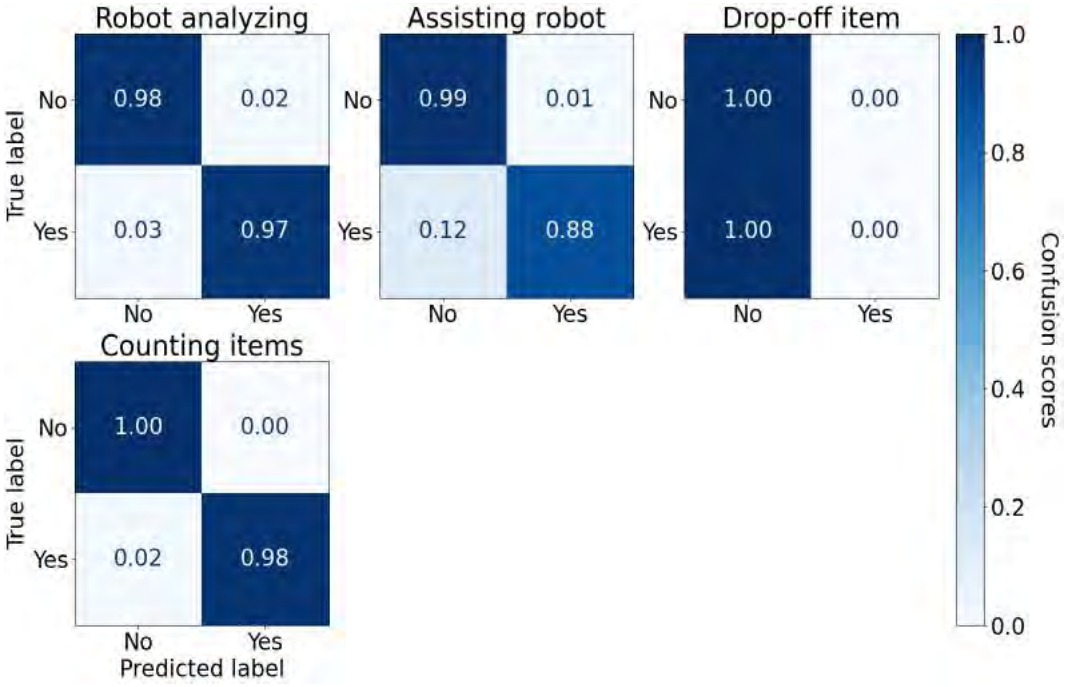


Figure B.11: Speech component's confusion matrices post GNN fusion algorithm's consolidation for the 1s, 3s, 10s, and 15s window sizes.

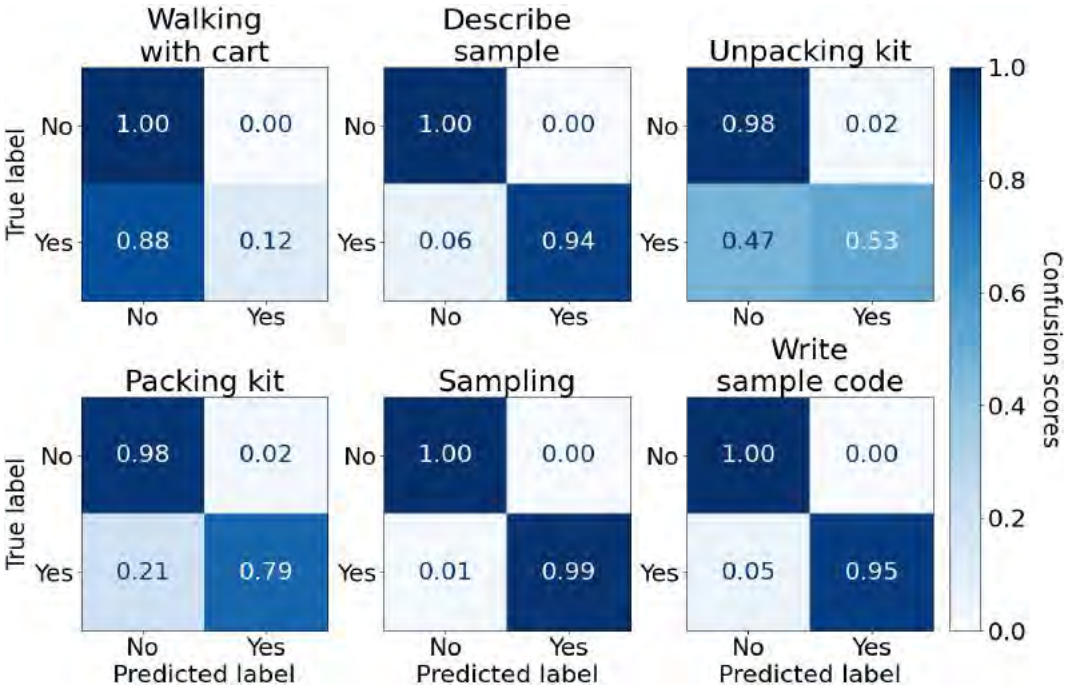
B.6 Composite and Concurrent Task Recognition

B.6.1 1s Window Size

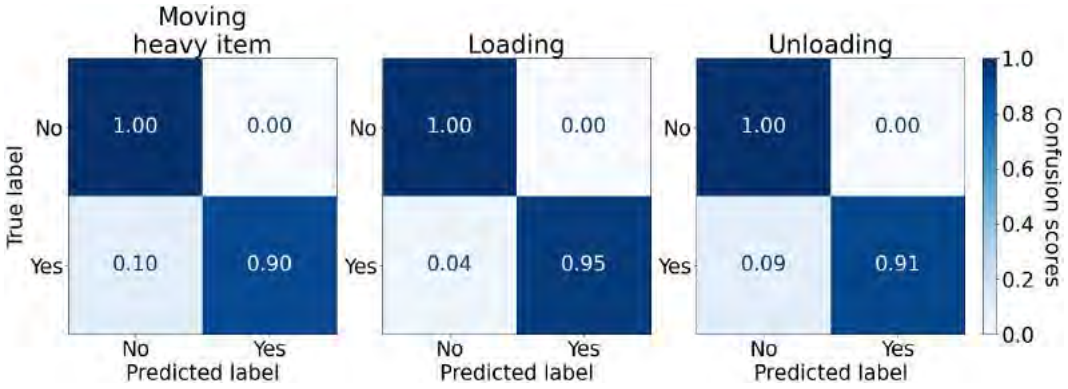
(a) Multi-label confusion matrices for the composite tasks that were shared across missions.



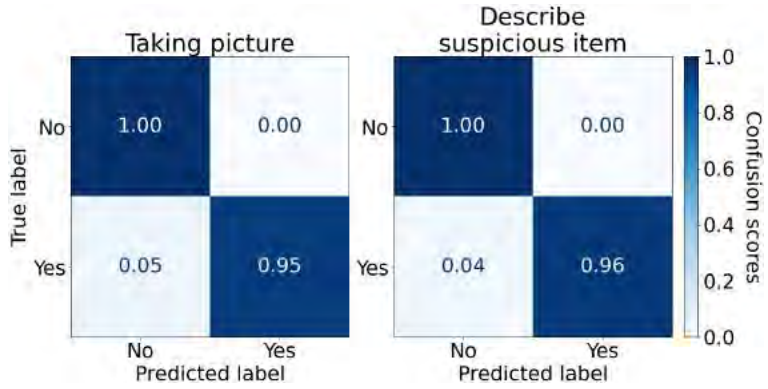
(b) Multi-label confusion matrices for the Pharmacy and Pawnshop missions' composite tasks.



(c) Multi-label confusion matrices for the Solid and Liquid sampling missions' composite tasks.



(d) Multi-label confusion matrices for the Debris mission's composite tasks.

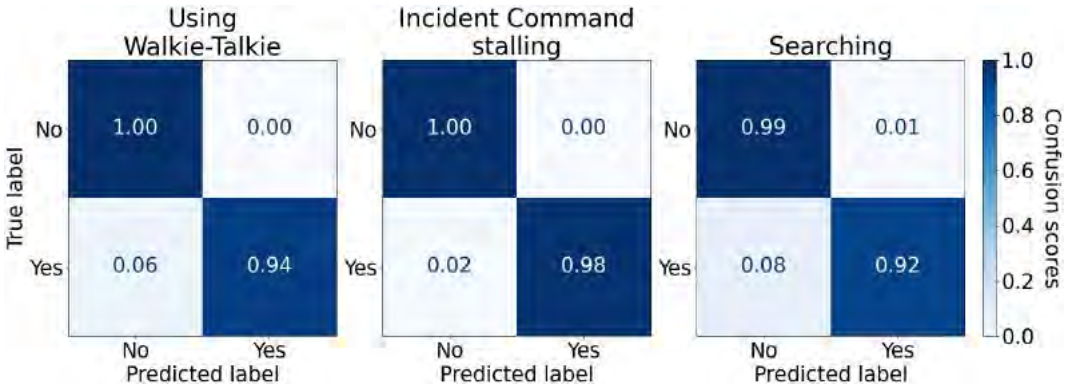


(e) Multi-label confusion matrices for the Search mission's composite tasks.

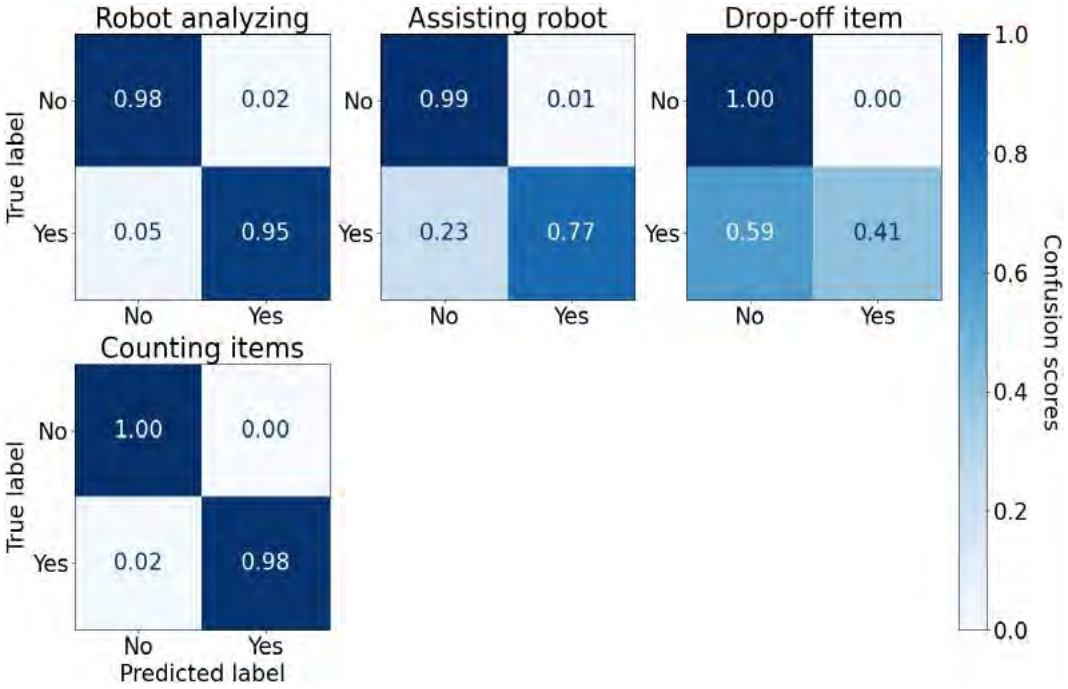
(f) Multi-label confusion matrices for the Secondary composite tasks.

Figure B.12: The TCN algorithm's 1s window size variant's multi-label confusion matrices grouped by mission and secondary tasks aggregated across participants.

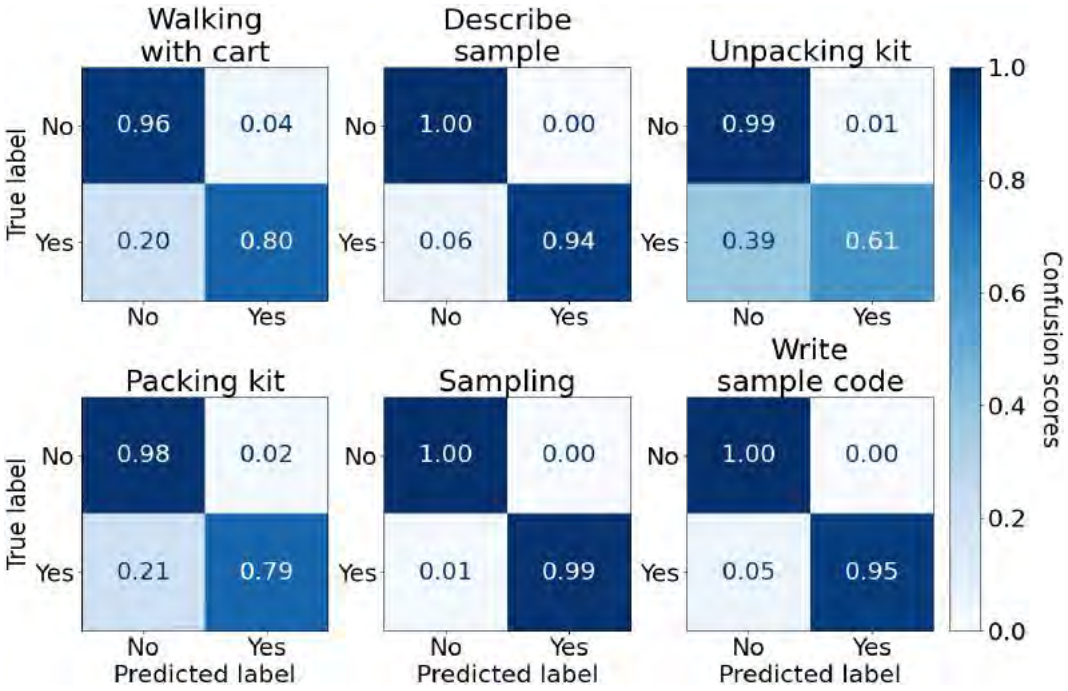
B.6.2 3s Window Size



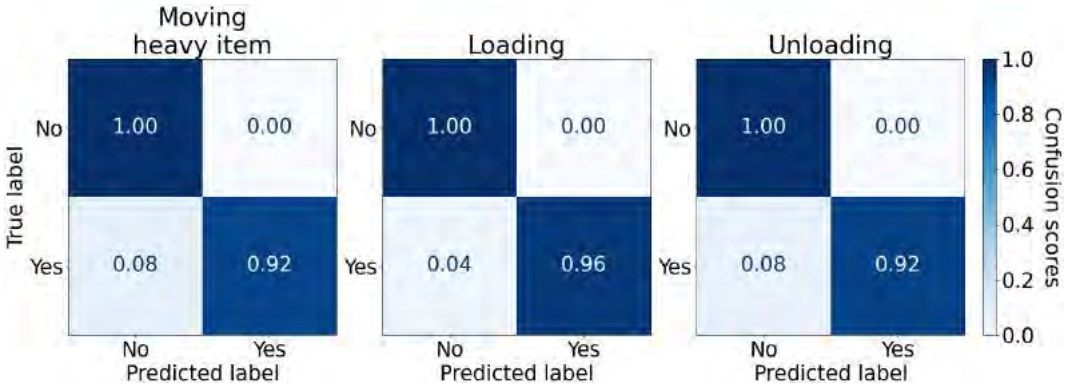
(a) Multi-label confusion matrices for the composite tasks that were shared across missions.



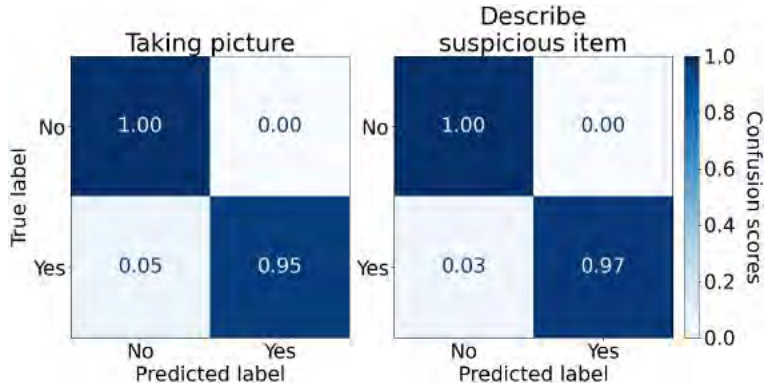
(b) Multi-label confusion matrices for the Pharmacy and Pawnshop missions' composite tasks.



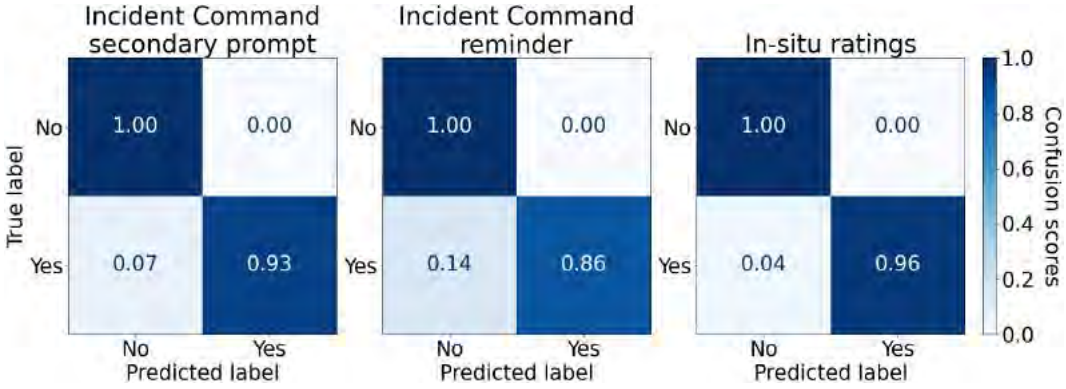
(c) Multi-label confusion matrices for the Solid and Liquid sampling missions' composite tasks.



(d) Multi-label confusion matrices for the Debris mission's composite tasks.



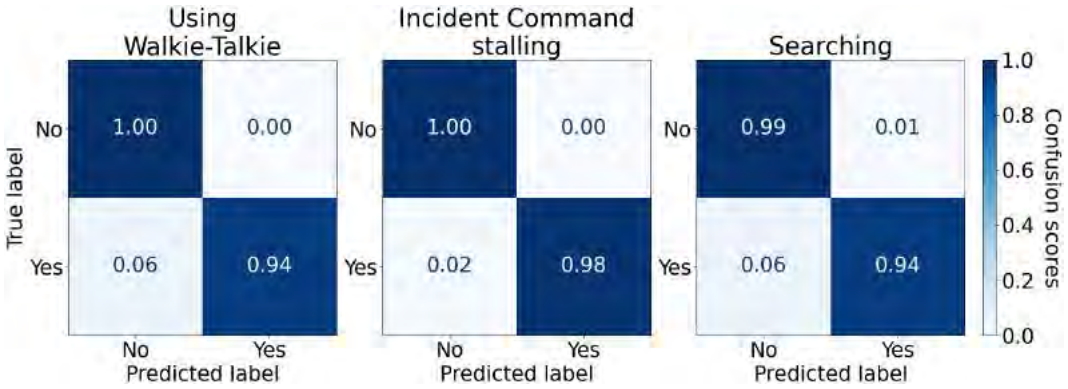
(e) Multi-label confusion matrices for the Search mission's composite tasks.



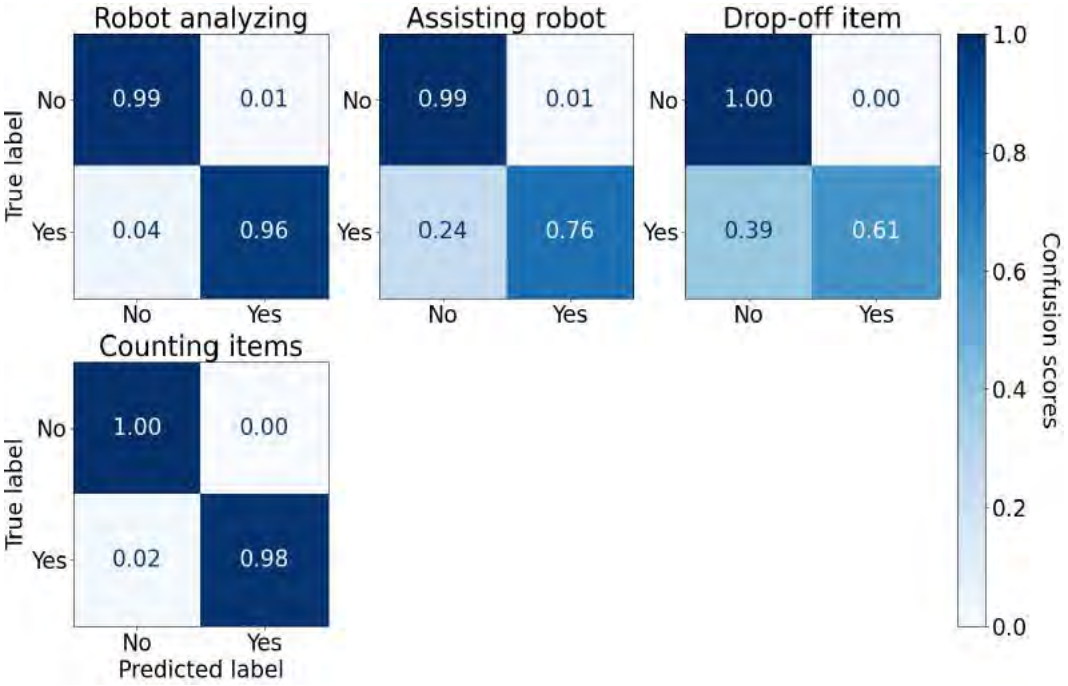
(f) Multi-label confusion matrices for the Secondary composite tasks.

Figure B.13: The TCN algorithm's 3s window size variant's multi-label confusion matrices grouped by mission and secondary tasks aggregated across participants.

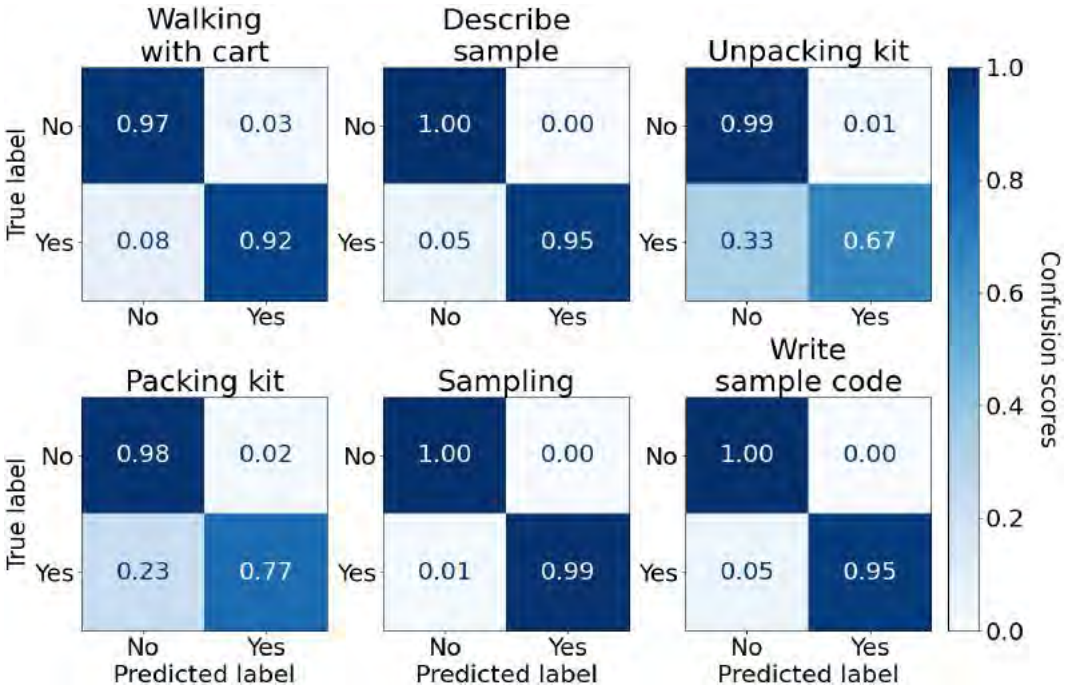
B.6.3 5s Window Size



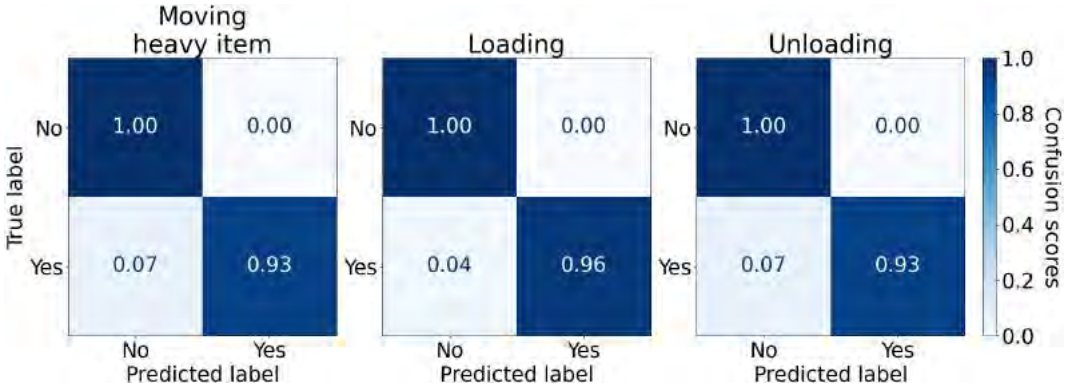
(a) Multi-label confusion matrices for the composite tasks that were shared across missions.



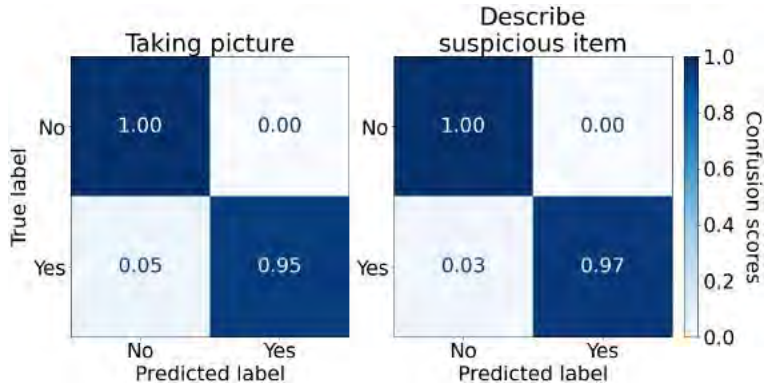
(b) Multi-label confusion matrices for the Pharmacy and Pawnshop missions' composite tasks.



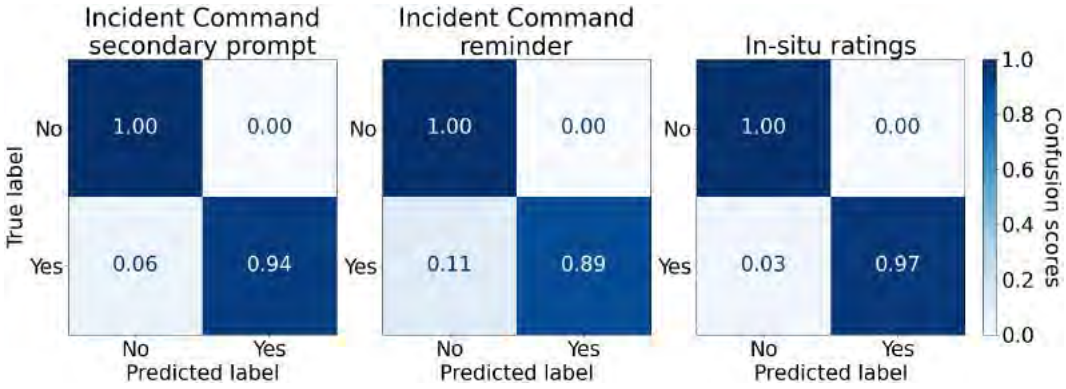
(c) Multi-label confusion matrices for the Solid and Liquid sampling missions' composite tasks.



(d) Multi-label confusion matrices for the Debris mission's composite tasks.



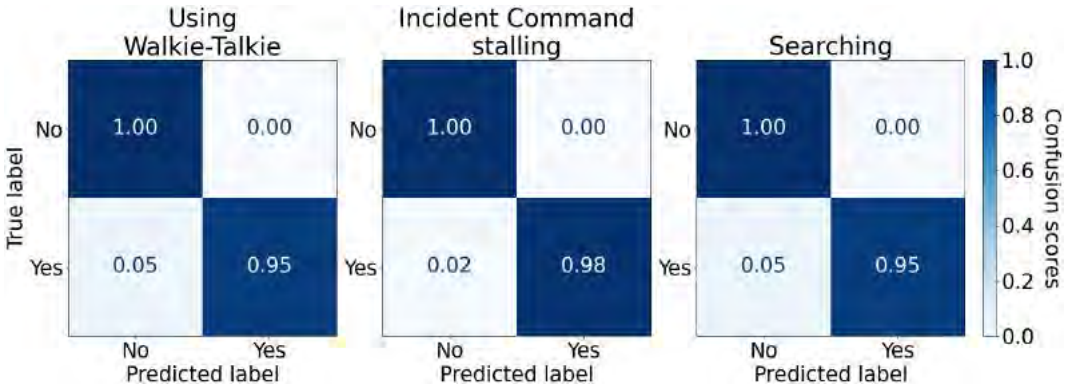
(e) Multi-label confusion matrices for the Search mission's composite tasks.



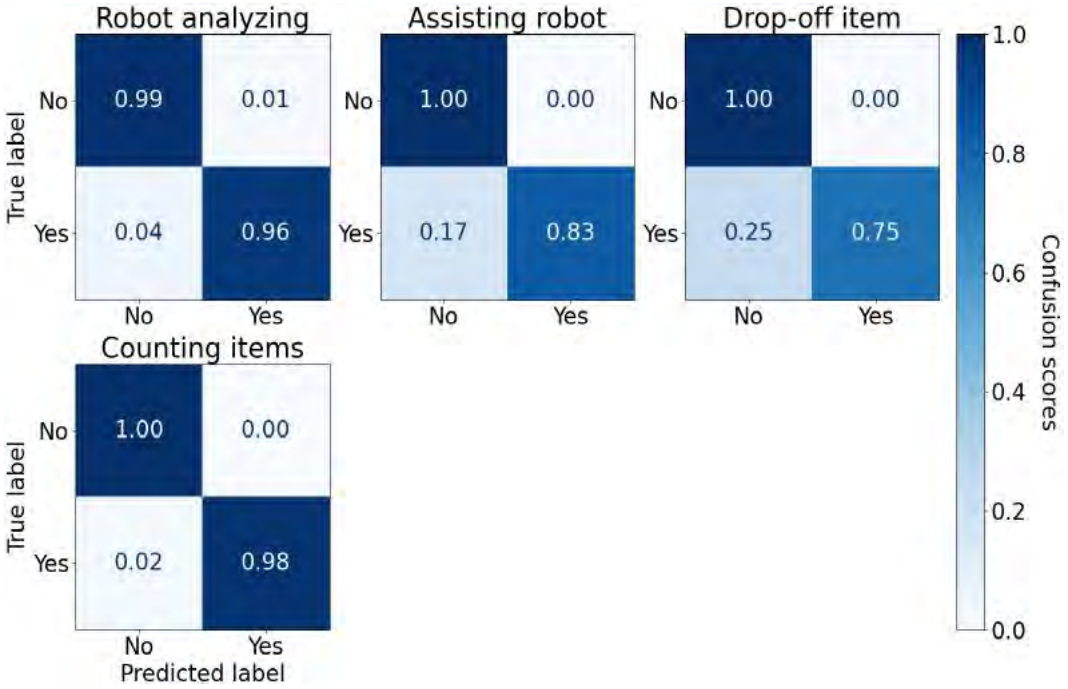
(f) Multi-label confusion matrices for the Secondary composite tasks.

Figure B.14: The TCN algorithm's 5s window size variant's multi-label confusion matrices grouped by mission and secondary tasks aggregated across participants.

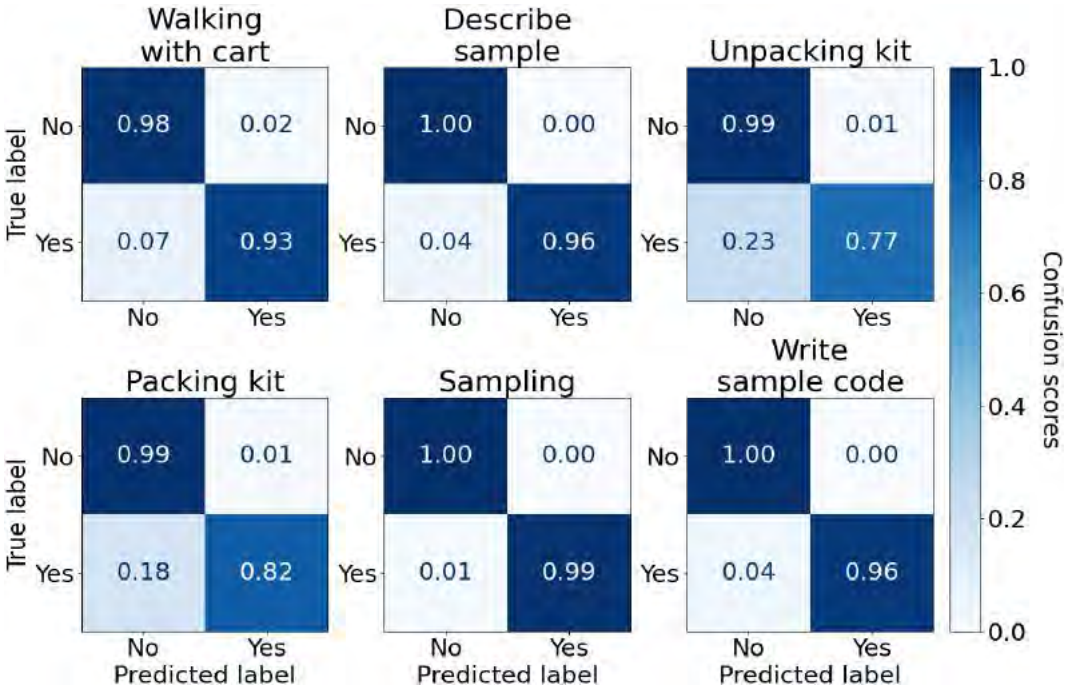
B.6.4 10s Window Size



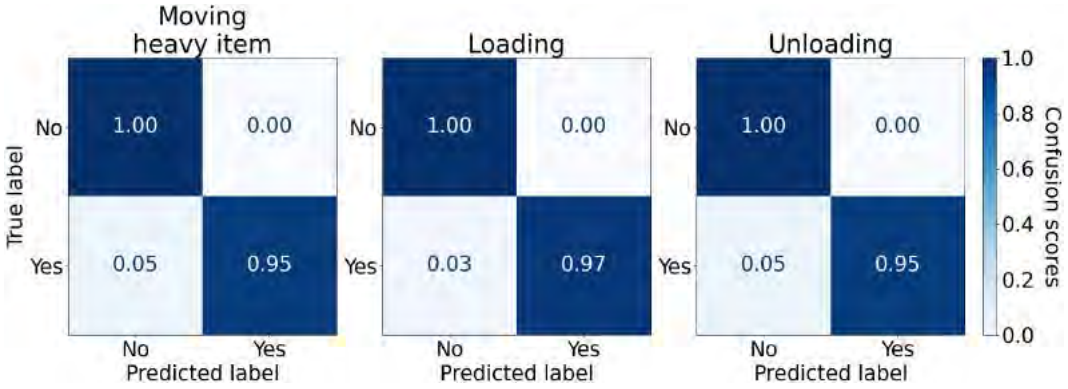
(a) Multi-label confusion matrices for the composite tasks that were shared across missions.



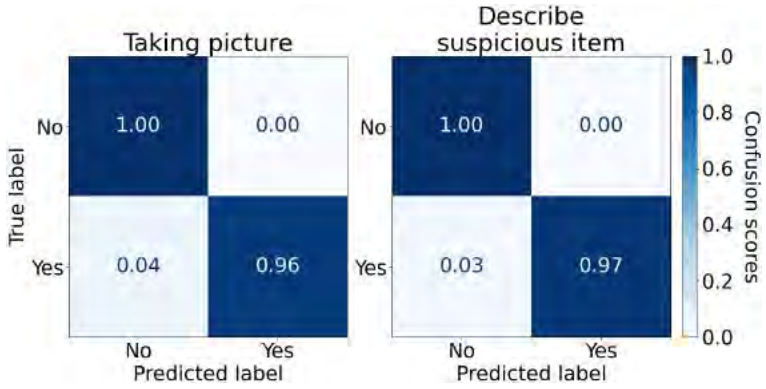
(b) Multi-label confusion matrices for the Pharmacy and Pawnshop missions' composite tasks.



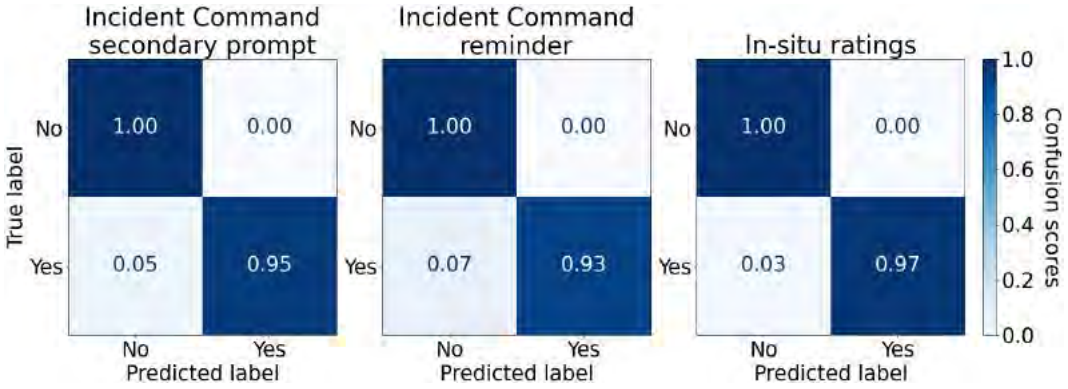
(c) Multi-label confusion matrices for the Solid and Liquid sampling missions' composite tasks.



(d) Multi-label confusion matrices for the Debris mission's composite tasks.



(e) Multi-label confusion matrices for the Search mission's composite tasks.



(f) Multi-label confusion matrices for the Secondary composite tasks.

Figure B.15: The TCN algorithm's 10s window size variant's multi-label confusion matrices grouped by mission and secondary tasks aggregated across participants.