A Multimodal Execution Monitor with Anomaly Classification for Robot-Assisted Feeding

Daehyung Park*, Hokeun Kim, Yuuna Hoshi, Zackory Erickson, Ariel Kapusta, and Charles C. Kemp

***Abstract*— Activities of daily living (ADLs) are important for quality of life. Robotic assistance offers the opportunity for people with disabilities to perform ADLs on their own. However, when a complex semi-autonomous system provides real-world assistance, occasional anomalies are likely to occur. Robots that can detect, classify and respond appropriately to common anomalies have the potential to provide more effective and safer assistance. We introduce a multimodal execution monitor to detect and classify anomalous executions when robots operate near humans. Our system builds on our past work on multimodal anomaly detection. Our new monitor classifies the type and cause of common anomalies using an artificial neural network. We implemented and evaluated our execution monitor in the context of robot-assisted feeding with a general-purpose mobile manipulator. In our evaluations, our monitor outperformed baseline methods from the literature. It succeeded in detecting 12 common anomalies from 8 able-bodied participants with 83% accuracy and classifying the types and causes of the detected anomalies with 90% and 81% accuracies, respectively. We then performed an in-home evaluation with Henry Evans, a person with severe quadriplegia. With our system, Henry successfully fed himself while the monitor detected, classified the types, and classified the causes of anomalies with 86%, 90%, and 54% accuracy, respectively.**

I. Introduction

Activities of daily living (ADLs), such as feeding and hygiene related tasks, are important for living independently and having a high quality of life [1]. Robotic assistance could help people with disabilities perform ADLs on their own. This notion is motivated by a number of examples, such as, robot-assisted shaving [2], dressing [3], and feeding [4]. Particularly in feeding, many specialized commercial systems are now available, such as the Bestic arm [5], Obi [6], and Mealtime partner [7]. Introducing greater autonomy into systems for robot-assisted feeding has the potential to improve their effectiveness. For example, visually tracking and autonomously moving utensils to the mouths of users could make the systems easier to use, simplify system setup, and enable a more diverse set of users to benefit from assistance. General-purpose robots also have the potential to provide feeding assistance along with other forms of assistance, instead of being single-purpose robots [8].

While greater autonomy and general-purpose robots have the potential to improve assistance, they also increase the complexity of robotic systems. This increase in complexity can lead to a higher likelihood of anomalies when providing assistance in the real world. While physically assisting a person with disabilities, anomalies could decrease system safety, effectiveness, and usability. Previously, we presented an execution monitor that uses multimodal sensing to detect anomalies [9]. In this paper, we focus on the problem of classifying and responding to common anomalies. We also present an evaluation of our robot-assisted feeding system with a person with severe disabilities in his own home. Our robot-assisted feeding system incorporates autonomy and makes use of a general-purpose mobile manipulator, a PR2.

Fig. 1: Henry Evans, a person with severe quadriplegia, successfully used our robot-assisted feeding system in his home to feed himself while our execution monitor was running.

Our execution monitor consists of a data-driven anomaly detector and classifier (Fig. 2). For the detector, we combine two multimodal anomaly detectors, referred to as HMM-D [9], that use multivariate hidden Markov models (HMMs) and dynamic thresholds to determine anomalies. Our anomaly classifier uses a multilayer perceptron (MLP) with selected input features extracted from HMMs, raw sensory signals, and a convolutional neural network (CNN). To use unexpected changes of signals, our system extracts conditional probabilities of each signal over others and then extract its temporal changes using temporal pyramid pooling as addressed in [10]. We also extract bottleneck features, the output of the CNN given an image. After detection, an MLP fuses these features and estimates the most probable class of the anomaly among 12 classes selected through fault tree analysis (see Fig. 3).

We evaluated the detection and identification performance of our execution monitor using a robot-assisted feeding dataset where a PR2 fed 8 able-bodied participants. We then evaluated our feeding and execution monitoring system with Henry Evans, a person with quadriplegia in California,

D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta, and C. C. Kemp are with Healthcare Robotics Lab, Georgia Institute of Technology, Atlanta, GA. *D. Park is the corresponding author deric.park@gatech.edu.

USA (see Fig. 1). The robot recorded haptic, auditory, kinematic, and visual data from a variety of sensors during the feeding task. Our execution monitor successfully detected and identified a variety of anomalous executions, such as the spoon missing the person's mouth, unexpected collisions during feeding, and a loud utterance from the care receiver or a nearby caregiver. Our method resulted in substantially higher classification accuracy when compared against six other baseline classification methods from the literature.

II. Related Work

Execution monitoring has been well-studied in robotics for detecting and classifying anomalous executions [11]. Bjreland introduced a monitoring system that detects, classifies, and corrects anomalous executions using a predictive model [12]. Pettersson introduced another monitoring system which detects and indicates anomalies in robot behaviors without restricting the detection method to a predictive model [13]. Unlike this previous work, our execution monitor observes the status of a task-relevant object and a person.

Anomaly detection has been investigated for various assistive devices. For example, Geravand and et al. introduced a fall detector that monitors force-torque data to predict and prevent falling during mobility assistance [14]. Colombo et al. showed environmental anomaly detection (e.g. wet floors, road block, or a change in environment) using a visual modality with a robotic walker, DALi [15]. We also introduced an anomaly detector that checks multimodal sensory signals to monitor robot assistance, such as robot-assisted feeding [9]. Our new system monitors more modalities (e.g., sound source direction, force on skin) and classifies anomalies. Unlike our old system, we have also successfully tested our new system with a person with disabilities.

Anomaly classification is also known as fault isolation or diagnosis [16], [17] and is part of the fault detection and isolation defined by IFAC SAFEPROCESS committee in 1993. As we discuss below, the classification has been used to determine the source of anomalies while running manipulators or mobile robots [18]. Based on Pettersson's classification [11], we can classify relevant work into three groups: causal analysis, expert systems, and data-driven classification. Causal analysis finds the cause of an anomaly based on the relationship between the fault and the cause (e.g., a signed directed graph [19]). Expert systems are widely applied in industry to isolate the cause of an anomaly using "IF THEN" rules or Fuzzy logic [20]. These two approaches often make use of extensive, detailed programming by domain experts. On the other hand, a data-driven approach is feasible if anomaly data are available. Several researchers have applied neural network-based classifiers for anomaly classification to robotic manipulation tasks, such as [21], [22]. Yamazaki et al. performed a database search for the closest anomaly type given a detected failure in a robot-assisted dressing task. [23]. We also use a data-driven classifier to fuse multimodal sensory data and classify anomalies of known classes.

Multimodal fusion and classification are also a closely related area. Ngiam et al. introduced a multimodal fusion

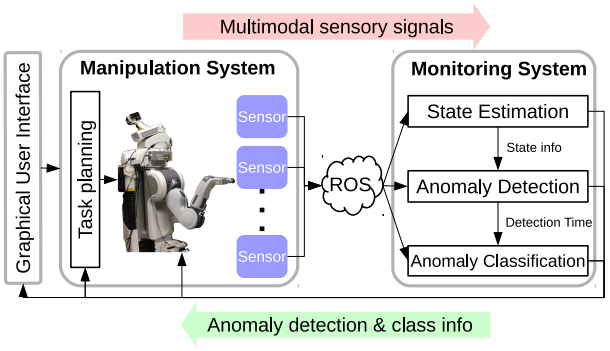

Fig. 2: Overview of the multimodal execution monitor.

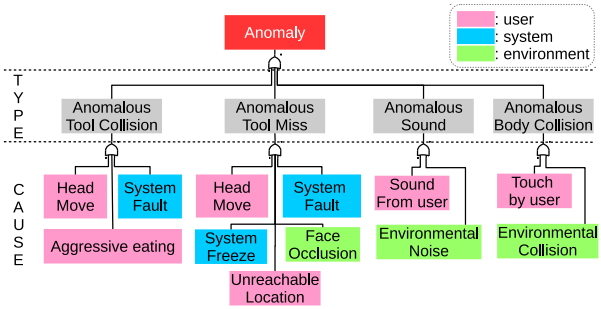

Fig. 3: Fault tree analysis for the robot-assisted feeding task.

method that generates a shared representation between modalities using a single network [24]. They discussed three different levels of fusions: early, intermediate, and late fusion. In this paper, we use late fusion with an MLP. Sung et al. showed a multimodal classification method that finds a desired trajectory from a feature space where one or more modalities are separately embedded [25]. We also embed multimodalities into a space after concatenating modalities.

III. Anomaly Detection

A. Anomaly Definition

The Oxford English Dictionary defines an anomaly as "something that deviates from what is standard, normal, or expected". As such, an anomaly can be recognized by several other terminologies: a fault, an outlier, or an unforeseen situation [26], [27]. Ogorodnikova classified the causes of anomalies into 3 groups: engineering, human, and environmental conditions [28]. Although robotic assistance might benefit from robots that are able to identify the cause of an anomaly, causal inference is difficult due to the complexity of real-world manipulation.

Instead, we focus on the detection and classification of representative anomalies that are more likely to occur during the direct task at hand. For robot-assisted feeding, we identified 12 anomalies through fault tree analysis which is a deductive analysis approach for resolving system hazards into their causes. Fig. 3 illustrates the results of our fault tree analysis. In the figure, depth 1 and 2 represent the types and causes of anomalies, respectively. The three colors used in depth 2 represent the causal groups from [28], [29].

B. Detection Framework

Our system detects anomalies with HMM-D detectors, which we introduced in [9]. HMM-D is a binary (one-

class) detector that learns a model from non-anomalous task executions and detects anomalies when the log-likelihood of a sequence of input signals is lower than a time-varying threshold. HMM-D dynamically changes the threshold depending on the progress of a current task execution. In this study, we used 25 hidden states for HMMs and 25 Gaussian radial basis functions for the threshold selection.

In contrast to our previous work, our new system uses two HMM-D anomaly detectors, each responsible for modeling a different set of sensory features. If either detector detects an anomaly, then the system detects an anomaly. In practice, we found that decomposing anomaly detection in this way resulted in improved system performance. The two detectors use the following sensory features:

- 1st Detector: *sound energy*, *1st joint torque*, *accumulated force*, and *spoon-mouth distance*
- 2nd Detector: *spoon speed*, *force*, *desired spoon displacement*, and *spoon-mouth distance*.

Before training each of the two HMM-D detectors, we first extract hand-engineered features from raw sensory signals after resampling and scaling, as described in [9]. Each detector uses four hand-engineered features that we found resulted in improved detection performance[1] based on cross-validation tests (see Section V-D).

We trained each HMM-D detector using the Baum-Welch algorithm as defined in the General Hidden Markov Model library (GHMM) (http://www.ghmm.org/). This training results in a set of HMM parameters λ which includes a transition probability matrix $\mathbf{A} \in \mathbb{R}^{25\times25}$ and Gaussian emission probabilities B. We then found parameters for the dynamically-changing threshold, which is defined by the expected log-likelihood $\hat{\mu}$ and its standard deviation $\hat{\sigma}$.

IV. Anomaly Classification

We introduce a supervised data-driven classifier that identifies representative anomalies from multimodal features.

A. Feature Extraction

Our monitoring system extracts two groups of multimodal features, temporal and convolutional features (see Fig. 4).

1) Temporal features: In our recent work under review [30], we found that particular types of anomalies tended to be more apparent in a subset of modalities. Consequently, the extent to which the feature from particular modalities is unexpected could be useful when classifying anomalies. It could be also helpful to determine the largest anomaly given multiple anomalies. To measure the unexpected change in a feature, our system estimates the likelihood of each feature with respect to the other features given by the HMMs used in anomaly detection. We refer to this estimate as a conditional likelihood. The conditional likelihood of a feature X_i is

$$P(X_i|X_{\mathbb{S}\setminus\{i\}},\lambda_{\mathbb{S}}) = \frac{P(X_{\mathbb{S}}|\lambda_{\mathbb{S}})}{P(X_{\mathbb{S}\setminus\{i\}}|\lambda_{\mathbb{S}})} \quad (1)$$

$$= \frac{P(X_{\mathbb{S}}|\lambda_{\mathbb{S}})}{P(X_{\mathbb{S}\setminus\{i\}}|\lambda_{\mathbb{S}\setminus\{i\}})}, \quad (2)$$

where $X_{\mathbb{S}}$ is a set of features in the HMM, $\lambda_{\mathbb{S}}$ is the HMM's parameters, and $\lambda_{\mathbb{S}\setminus\{i\}}$ is a part of λ that excludes the elements related to X_i.

In this paper, we use multivariate Gaussian emissions in HMMs. The marginal distribution of a multivariate Gaussian distribution is a Gaussian: $P(x_1,...,x_n) = \mathcal{N}(\mu_{1:n}, \mathbf{\Sigma}_{1:n})$ given $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$ [31]. Thus, the denominator of Eq. (1) can be converted to the denominator of Eq. (2). For computational convenience, we use the logarithm of (2), i.e., $l_i = logP(X_i|X_{\mathbb{S}\setminus\{i\}},\lambda_{\mathbb{S}})$. At each time step t, we extract a total of 8 conditional log-likelihoods from two 4-dimensional HMMs and concatenate these to create a feature vector $\mathbf{v}_t$ in $\mathbb{R}^8$,

$$\mathbf{v}_t = \{l_1^1, l_2^1, l_3^1, l_4^1, l_1^2, l_2^2, l_3^2, l_4^2\}, \quad (3)$$

where the superscript shows the identity of the HMM.

To represent the temporal change of this feature vector, we perform 3 levels (1-4-8) of temporal pyramid pooling [10] given the last 8 time steps of feature vectors $\mathbf{v}_{[t-8\Delta t:t]}$. Fig. 5 shows this pooling process for which we partition the feature vectors into 3 cells and pool the minimum value per feature (i.e., conditional log-likelihood of each feature). The pooling from 3 cells give 3 vectors, $\mathbf{v}^1$, $\mathbf{v}^4$, and $\mathbf{v}^8$. We then concatenate the vectors to form a single vector $[\mathbf{v}^1, \mathbf{v}^4, \mathbf{v}^8]$ in $\mathbb{R}^{24}$. For this paper, we chose minimum pooling since a drop in likelihood would indicate an unexpected change in the features.

In addition to the conditional log-likelihood features, we include 7 additional features that can be useful in identifying the cause of an anomaly. These include: *frontal sound amplitude*, *sound source direction (azimuth angle)*[2], *x-direction force*, *y-direction force*, *contact force on the whole-arm tactile sensing skin*, *distance between the robot's torso and the person's mouth*, and *spoon-mouth angular difference*. For each feature, we pool a minimal or maximal value over the last 20 time steps, which we refer to as min-max pooling (see Algorithm 1). This pooling enables us to extract the unexpected change of each feature since the 7 features are usually static in non-anomalous executions. We empirically decided on 20 time steps for the pooled features, $\mathbf{v}^e$ in $\mathbb{R}^7$, to include only recent feature changes. The final feature vector $\mathbf{V}_T = [\mathbf{v}^1, \mathbf{v}^4, \mathbf{v}^8, \mathbf{v}^e]$ is of length 31 ($= 8 \times 3 + 7$).

Algorithm 1: Min-max Pooling

Data: A sequence of observations, $\mathbf{x}$
Result: Pooled value
if $|min(\mathbf{x})| > |max(\mathbf{x})|$ **then**
  return min($\mathbf{x}$) ;
else
  return max($\mathbf{x}$) ;

2) Convolutional features: The interpretation of visual information can help to better classify anomalies during feeding. Our algorithm extracts the output of a convolutional

[1] Area under curve (AUC) for a receiver operating characteristic curve

[2] We localize the source of sound using interaural time differences [32].

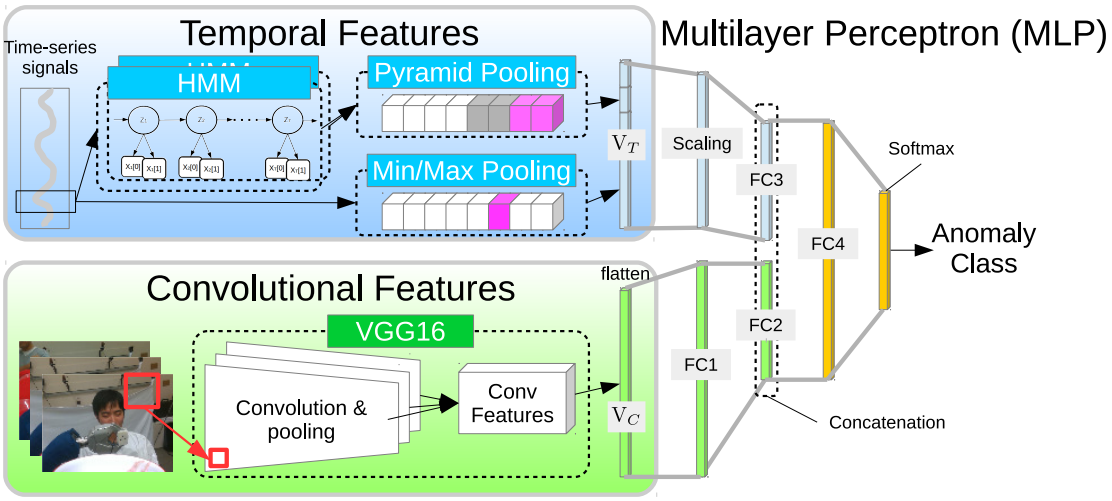

Fig. 4: Illustration of our anomaly classification network. The MLP outputs the most probable class of an anomaly using temporal and convolutional features. The size of the FC1-FC4 layers are 1024, 128, 128, and 256, respectively.

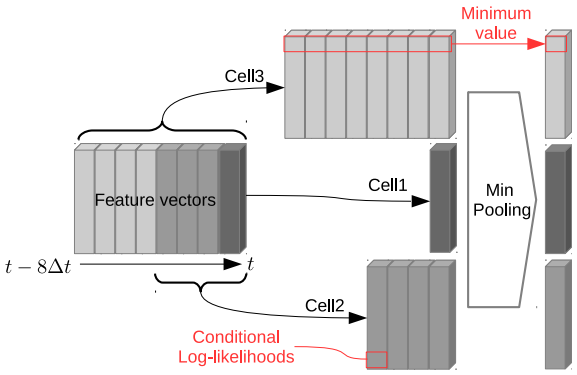

Fig. 5: Illustration of the temporal pyramid pooling process. The boxes show the 3-level temporal partition of a sequence. The output feature vector is of size $8 \times 3 = 24$.

neural network (CNN) as bottleneck features [33], for images collected by the PR2. The features can represent the existence of objects around the work space. In this paper, we use the VGG16 CNN model which has been trained on the ImageNet dataset with 1,000 classes [34]. When an anomaly is detected, our network takes as input a 224x224 RGB image of the person captured from the wrist-mounted camera. Note that the PR2 always positions the camera in front of the person to ensure his or her face is in the captured image (see captured images in Fig. 4). We then flatten output features ($\in \mathbb{R}^{512\times7\times7}$) to a vector $\mathbf{V}_C$ which is input into a multilayer perceptron (MLP) (see Fig. 4). In this work, we used Keras, a deep learning library [35], with the Tensorflow back end to load the VGG16 network and extract bottleneck features.

B. Multilayer Perceptron (MLP) Classification

The execution monitor uses a multilayer perceptron (MLP) to classify the types and causes of anomalies given the multimodal temporal and convolutional features. An MLP is a feedforward artificial neural network with fully-connected layers. The right side of Fig. 4 presents our MLP structure. Our MLP consists of three fully-connected layers with rectified linear units (ReLU). A softmax function is applied to the final layer for multiclass classification.

Before fusing the temporal and convolutional features, we feed the convolutional features through the first fully-connected layer of our MLP. We then concatenate the temporal features ($\in \mathbb{R}^{128}$) and the first layer's output ($\in \mathbb{R}^{128}$) to a vector ($\in \mathbb{R}^{256}$) similar to common CNN-LSTM models [36]. Fig. 6 shows the distribution of the concatenated multimodal features using data from trials with 8 able-bodied participants. We display the first two principal components from a principal component analysis (PCA).

To train this network, we used only positive (anomalous) feeding trials. It is difficult to label exactly when an anomaly has started to occur and the anomaly detector's sensitivity can influence the timing of detections. To account for this, we first set the thresholds of the HMM-D detectors to maximize detection accuracy given a training dataset. We then collected the temporal and convolutional features when the system first detected an anomaly. We also collected additional feature sets over 10 time steps before and after the time at which the detection occurred to account for variation in the timing of anomaly detections. Before concatenating the features at each time step, we performed feature-wise scaling for the temporal features to have zero mean and unit variance. We trained the MLP classifier initialized with uniformly distributed random weights using a stochastic gradient descent (SGD) optimizer with RMSProp. We then fine tuned it with a conventional SGD optimizer. To avoid overfitting, we added dropout and L2-regularization to each layer [37].

During real time experiments, our system extracts features and performs anomaly classification only after an anomaly has been detected.

V. Evaluation

We evaluated our multimodal execution monitor with a robot-assisted feeding system that performs scooping and delivers a spoon of yogurt to the mouth of participants. We conducted our evaluations with approval from the Georgia Tech Institutional Review Board (IRB).

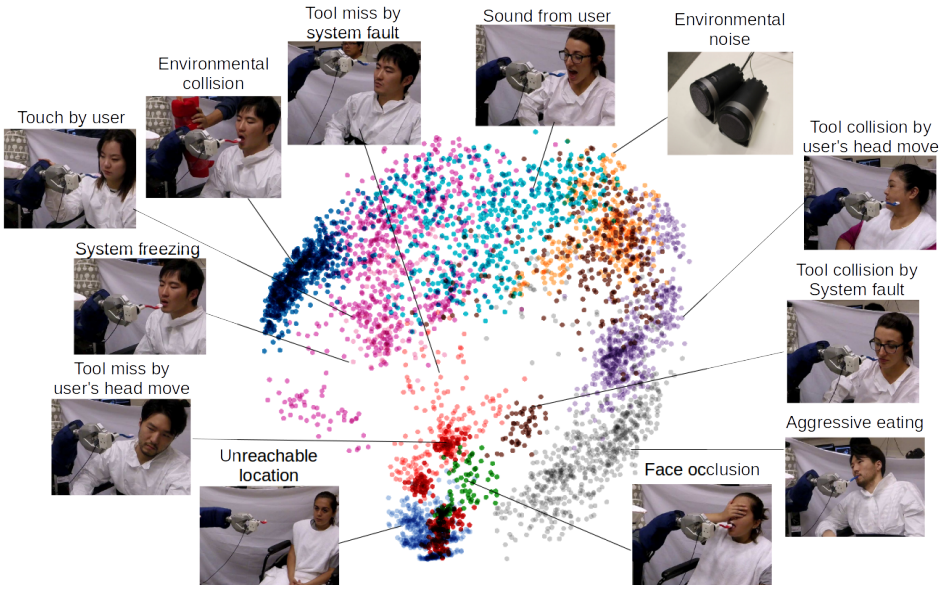

Fig. 6: Distribution of multimodal features after concatenating the outputs of the FC2 and FC3 layers described in Fig. 4. We plot the features on the first two principal components using principal component analysis (PCA). The images come from a separate camera used to record the trials, not the camera used by the execution monitor.

A. Instrumental Setup

Our robot-assisted feeding system uses a PR2 from Willow Garage, which is a general-purpose mobile manipulator. The PR2 consists of an omni-directional mobile base and two 7-DOF back-drivable arms with powered grippers. We run a 1 kHz low-level PID controller with low gains and a 50 Hz mid-level model predictive controller from [38] without haptic feedback. We designed 3D-printed handles so the PR2 can grip both a spoon and a bowl. We also affixed a spill guard and bars for wiping the spoon to the bowl. After scooping yogurt, the robot can drag the spoon across the bars to clean off yogurt from the bottom of the spoon (see Fig. 7).

We mounted multiple sensors on the robot for multimodal sensing during feeding assistance. We mounted a force/torque sensor (ATI Nano25) between the handle and the spoon to measure the forces and torques applied to the spoon by the user at 1 kHz. To estimate the location of a user's mouth, we mounted an RGB-D camera (Intel SR300) on the right arm's wrist (see Fig. 1). We also use the Intel SR300's 2-channel microphone array to measure and localize sounds. To sense collisions with the robot's body, we covered the robot's left arm with fabric-based whole-arm tactile sensors introduced in [39]. These tactile sensors provide both contact locations and forces. Our monitoring system only uses the sum of all the estimated force magnitudes from the tactile sensors.

B. Robot-Assisted Feeding System

Our robot-assisted feeding system performs three autonomous subtasks (see Fig. 8). A user is able to command the robot to perform the 'scooping', 'clean spoon,' and 'feeding' subtasks using a web-based graphical user interface (GUI). Given the 'scooping' command, a PR2 estimates the location of the bowl held by its right arm and then scoops a spoon of yogurt using predefined motions. To avoid spilling

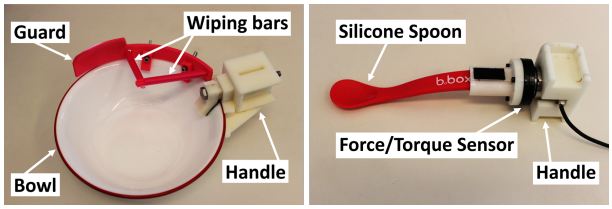

Fig. 7: **Left:** A bowl with an attached handle, guard, and wiping bars to avoid spilling food. **Right:** A tool for feeding that has a flexible silicone spoon and force-torque sensor.

yogurt, the user can then command the 'clean spoon' subtask, which involves the PR2 dragging the back of the spoon over the wiping bars. Given the 'feeding' command, the robot estimates the user's mouth location using the SR300 camera and then moves the spoon to the user's mouth, inserts it, and retracts it. At any time during the feeding task, the user can stop the robot by clicking anywhere on the screen and then resume feeding by re-executing the previous subtask.

C. Simulated Anomalies

We asked able-bodied participants to produce 12 representative anomalies for the evaluations of our feeding and execution monitoring system. Fig. 6 shows the anomalies produced by the participants. Prior to participants producing simulated anomalies, we showed a demonstration video of several possible anomalies and instructed them on what to do. We then encouraged participants to produce any of the anomalies at any time with any variation. When we collected data with Henry Evans, a person with severe quadriplegia, we asked his wife and primary caregiver, Jane Evans, to produce some of the anomalies after seeing the demonstration video.

D. Evaluation Process

We first evaluated our monitoring system with 8 able-bodied participants. We recruited able-bodied participants—3

Fig. 8: An image sequence of the entire scooping and feeding process with Henry Evans in the living room of his home.

males and 5 females—whose ages ranged from 19 to 35. They were all novice users who did not have any experience with our feeding system. Each participant performed 20 non-anomalous and 24 anomalous executions over a total of 1.5 hours in a closed experiment room. The 24 anomalous executions consisted of all 12 anomalous cases being recorded two times. In total, we collected data from 160 non-anomalous and 192 anomalous feeding trials. During non-anomalous executions, we asked participants not to move their upper bodies and arms to approximate a lack of movement due to disabilities. To quantify the performance of our system, we performed leave-one-person-out cross-validation by training our execution monitor with 7 participants' data and testing the monitor with the 1 participant remaining.

We also included 508 hand-labeled images, which includes 12 anomalies, from 2 extra participants—average 33 years old of 2 females—who were not included in the above dataset but were recorded when piloting the experiment. We also performed data augmentation in which we added Gaussian random noise to individual signals for the temporal features and random rotations, translations, and magnifications of images for the convolutional features. We trained and tested our execution monitor using an Amazon EC2 server with 4 cores, 61GB of memory, and an NVIDIA K80 GPU.

We also tested our system with Henry Evans at his home in California, USA over a span of 4 days. This was our first test of our system with a person with disabilities. Our lab has an ongoing long-term collaboration with Henry Evans and his wife and primary caregiver, Jane Evans. Henry Evans is severely impaired, but he can move his head and a finger sufficiently well to operate our system's web-based GUI using an off-the-shelf head tracker and a mouse button. Henry is unable to speak and has difficulty eating. He frequently eats yogurt, which he specifically requested when we initiated our work on robot-assisted feeding. On the first day, we began with safety training and then allowed Henry to practice with the feeding system until he became comfortable using it. Henry then performed 20 non-anomalous feeding trials during which he was able to successfully eat yogurt each time. For the following three days, he participated in 5 sessions during which we used the anomaly detection and classification systems trained on the data from the 8 able-bodied participants. In each session, Henry performed 10 non-anomalous and 12 anomalous yogurt feeding trials in random order for about 1 hour in total. A caregiver, Jane Evans, produced 'sound by user' and 'touch by user' for him.

E. Baseline Methods for Anomaly Classification

Our proposed method consists of an MLP with both temporal and convolutional features, or MLP(T+C). Below we present six alternative methods for performing anomaly classification. We compare each of these methods to our approach in Section VI. Note that we ran each classification method with the same anomaly detector.

- Random: This method randomly determines the type and cause of anomalies.
- SVM(R): A support vector machine (SVM) classifier with a radial basis kernel. Type and cause of anomalies are determined with raw data used for temporal features.
- SVM(H): The same SVM structure with histogram of oriented gradients (HOG) features extracted from images.
- SVM(T): The same SVM structure with temporal features.
- MLP(T): MLP with only temporal features.
- MLP(C): MLP with only convolutional features.

VI. Results

We first investigated how the use of multimodal signals helps to classify anomalies. Fig. 9 shows the distribution of multimodal signals observed from the robot-assisted feeding tasks with 8 able-bodied participants. The blue region shows the mean and standard deviations of 7 features from 160 non-anomalous feeding executions. We can observe clear patterns that were subsequently used to train the HMMs for anomaly detection. The red curves show an anomalous execution in which a spoon collided with a user's mouth due to a system fault. We can observe a short bump in 'force on spoon' from the collision around 1.5s. The unexpected change may be sufficient to detect the anomaly, but it is difficult to estimate its cause among the other causes. Instead, as we can see in Fig. 6, the use of multimodal sensory signals helps separate anomalous events into different regions.

We also evaluated the overall effectiveness, or accuracy, of our execution monitoring system. To do so, we used the feeding data from the 8 able-bodied participants and performed leave-one-person-out cross validation. Our anomaly detector achieved 83.27% accuracy in detecting anomalous trials throughout all 352 feeding trials. We then used all anomalous data from the 8 able-bodied participants to train and test our anomaly classifier. Fig. 10 shows the comparison of our proposed method against the 6 baseline methods. Our proposed method had a classification accuracy of 81.37% which was 5% higher ($p = 0.0097$ from t test) than the next best method, SVM(T). Compared to the MLP(T), our proposed method resulted in 17% higher accuracy due to the inclusion of the convolutional features.

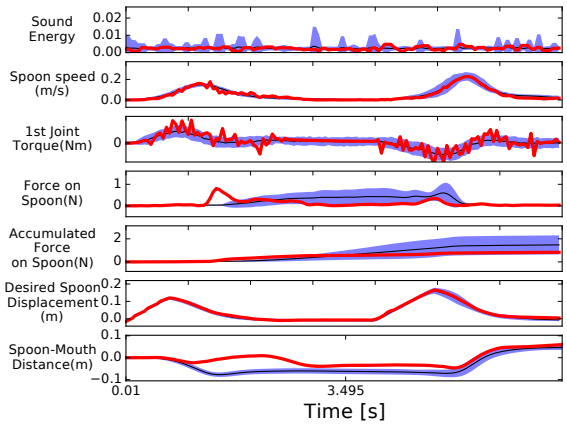

Fig. 9: Multimodal sensory signals used in the anomaly detection. Blue regions show their mean and standard deviation from non-anomalous feeding executions. Red curves show the signals from an anomalous feeding event: spoon collision due to a system fault at 1.5s.

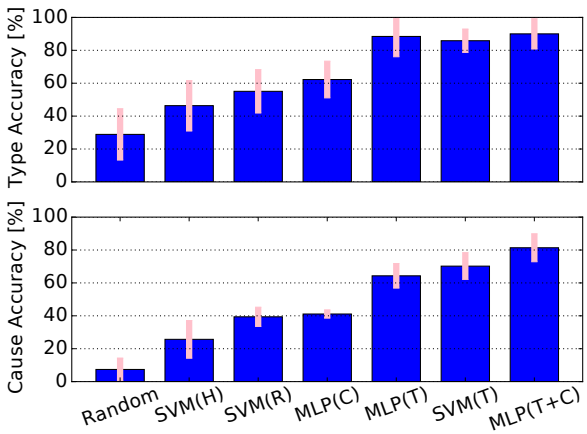

Fig. 10: Comparison of classification methods on 8 able-bodied participants' feeding data. The two bar charts show the detection accuracies with respect to the type and cause of anomalies described in Fig. 3. The symbols in parentheses indicate the type of input features (R=Raw signal data, C=convolutional features, H=HOG features, T=temporal features).

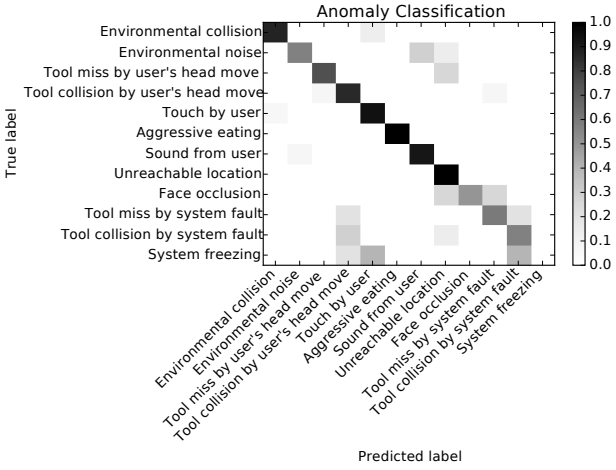

Fig. 11: Confusion matrix from the proposed anomaly classifier applied to data from 8 able-bodied participant using leave-one-person-out cross validation.

TABLE I: Five-point Likert type questionnaire items. The last column provides answers for strongly disagree (sd), disagree (d), neither (n), agree (a), and strongly agree (sa).

Question	answer
The system successfully accomplished tasks.	sa
I felt safe while using the system.	sa
The system was simple and easy to use.	sa
The anomaly detection helped me feel more safe.	a

Fig. 11 shows a confusion matrix for our classifier's performance with respect to the 12 known anomalies. Our proposed method successfully determined the causes of most anomalies except for the 'system freezing' event. In this work, we reproduced 'system freezing' by randomly killing the model predictive controller process while the robot was moving its arm to a location. However, our classification success for this event was limited by the fact that our system only successfully detected the anomaly 5 out of 16 times in training. This may be due to the lack of effective features, since no signal changes while in the freeze status, except 'desired spoon displacement' and our method did not monitor sensory streams related to the internal state of the robot.

In our first test with Henry Evans, he successfully fed himself with the robot for all 20 consecutive trials. That is, he ate 20 scoops of yogurt produced by Chobani, LLC. We then evaluated the execution monitor through 5 additional sessions, each of which included 10 non-anomalous and 12 different causes of anomalous executions in random order, for a total of 50 non-anomalous and 60 anomalous executions. Our monitoring system achieved 86.36% detection accuracy over all 110 executions with Henry. Our system also successfully detected two unintentional real anomalies caused by researcher mistakes and camera faults in the new environment. Our classifier successfully determined the types of anomalies with 90.51% accuracy (i.e., 'tool collision,' 'tool miss,' 'sound,' or 'body collision'). However, it classified the specific causes of anomalies with only 53.44% accuracy, which is roughly 30% lower than the cross-validation accuracy in our lab environment. In this evaluation, we did not train on any data collected from Henry.

Notably, our results with Henry Evans only used training data from able-bodied users in a controlled laboratory setting. The system's overall performance generalized reasonably well given that we conducted this test in Henry's home and that his impairments influence the way he eats. We expect that training on user-specific data and on more data with greater variation could improve results for in-home use.

At the end of our evaluation, we asked Henry to fill out a survey with 22 questions (five-point Likert type questionnaire items) based on [40], and 2 open-ended questions. Table I provides the questionnaire results that we found most informative. As can be seen from the table, Henry reported that he found the system to be effective, safe, and easy to use. His responses from the 22 questions indicated that the anomaly detection function positively contributed to his experience of using the robot, helping him feel safer, and effectively alerting him of problems. In an email following

the experiment, Henry also recommended several ways to improve the system, such as increasing the rate at which it feeds yogurt and giving the user the ability to finely adjust where the spoon moves with respect to the mouth.

VII. Conclusion

We introduced a multimodal execution monitor for a robot-assisted feeding system. The feeding system employs a general-purpose mobile manipulator (a PR2 robot) and provides a high-level web-based interface for people with disabilities. In a test with able-bodied participants, our execution monitor successfully detected and classified anomalies while outperforming 6 baseline methods. We also found multimodal features beneficial for classifying the causes of anomalies. In addition, we evaluated our system in the home of Henry Evans, a person with severe quadriplegia. Henry successfully used the system to feed himself, and the system detected and classified anomalies.

***Acknowledgment**: We thank Jane Evans and Youkeun Kim for their assistance throughout this project. This work was supported in part by NSF Awards IIS-1150157, IIS-1514258, NIDILRR grant 90RE5016-01-00 via RERC TechSAge, and a Google Faculty Research Award.*

References

[1] J. M. Wiener, R. J. Hanley, R. Clark, and J. F. Van Nostrand, "Measuring the activities of daily living: Comparisons across national surveys," *Journal of Gerontology*, vol. 45, no. 6, pp. S229–S237, 1990.

[2] K. P. Hawkins, P. M. Grice, T. L. Chen, C.-H. King, and C. C. Kemp, "Assistive mobile manipulation for self-care tasks around the head," in *Computational Intelligence in Robotic Rehabilitation and Assistive Technologies (CIR2AT), 2014 IEEE Symposium on*, IEEE, 2014.

[3] A. Kapusta, W. Yu, T. Bhattacharjee, C. K. Liu, G. Turk, and C. C. Kemp, "Data-driven haptic perception for robot-assisted dressing," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, IEEE, 2016.

[4] S. Schrer, I. Killmann, B. Frank, M. Vlker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6482–6487, May 2015.

[5] Camanio Care AB, "*Bestic, Increase your mealtime independence*," 2017. `http://www.camanio.com/` [Accessed: 2017-07-15].

[6] Eclipse Automation, "Obi," 2017. `https://meetobi.com/` [Accessed: 2017-07-15].

[7] Mealtime Partners, "*Specializing in Assistive Dining and Drinking Equipment*," 2017. `http://www.mealtimepartners.com/` [Accessed: 2017-07-15].

[8] D. Park, Y. K. Kim, Z. Erickson, and C. C. Kemp, "Towards assistive feeding with a general-purpose mobile manipulator," in *IEEE International Conference on Robotics and Automation - workshop on Human-Robot Interfaces for Enhanced Physical Interactions*, 2016.

[9] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *Robotics and Automation, 2016. ICRA'16. IEEE International Conference on*, IEEE, 2016.

[10] M. Madry, L. Bo, D. Kragic, and D. Fox, "St-hmp: Unsupervised spatio-temporal feature learning for tactile data," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014.

[11] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[12] M. Bjäreland, "Model-based execution monitoring," in *Linköping Studies in Science and Technology, Dissertation*, Citeseer, 2001.

[13] O. Pettersson, L. Karlsson, and A. Saffiotti, *Model-free execution monitoring in behavior-based mobile robotics*. Örebro universitetsbibliotek, 2004.

[14] M. Geravand, W. Rampeltshammer, and A. Peer, "Control of mobility assistive robot for human fall prevention," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Aug 2015.

[15] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards, "Efficient customisable dynamic motion planning for assistive robots in complex human environments," *Journal of ambient intelligence and smart environments*, vol. 7, no. 5, pp. 617–634, 2015.

[16] R. Isermann, "Supervision, fault-detection and fault-diagnosis methodsan introduction," *Control engineering practice*, vol. 5, no. 5, 1997.

[17] F. Caccavale and L. Villani, *Fault Diagnosis and Fault Tolerance for Mechatronic Systems: Recent Advances*, vol. 1. Springer Science & Business Media, 2002.

[18] R. Muradore and P. Fiorini, "A pls-based statistical approach for fault detection and isolation of robotic manipulators," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3167–3175, 2012.

[19] L. H. Chiang, R. D. Braatz, and E. L. Russell, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2001.

[20] P. Jackson, "Introduction to expert systems," 1986.

[21] R. Tinós and M. H. Terra, "Fault detection and isolation in robotic manipulators using a multilayer perceptron and a rbf network trained by the kohonens self-organizing map," *Rev Soc Bras Autom Contr Autom*, vol. 12, no. 1, pp. 11–18, 2001.

[22] T. Yüksel and A. Sezgin, "Two fault detection and isolation schemes for robot manipulators using soft computing techniques," *Applied Soft Computing*, vol. 10, no. 1, pp. 125–134, 2010.

[23] K. Yamazaki, R. Oya, K. Nagahama, K. Okada, and M. Inaba, "Bottom dressing by a life-sized humanoid robot provided failure detection and recovery functions," in *2014 IEEE/SICE International Symposium on System Integration*, pp. 564–570, Dec 2014.

[24] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 689–696, 2011.

[25] J. Sung, S. H. Jin, I. Lenz, and A. Saxena, "Robobarista: Learning to manipulate novel objects via deep multimodal embedding," *arXiv preprint arXiv:1601.02705*, 2016.

[26] A. Bouguerra, L. Karlsson, and A. Saffiotti, "Monitoring the execution of robot plans using semantic knowledge," *Robotics and autonomous systems*, vol. 56, no. 11, pp. 942–954, 2008.

[27] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, pp. 1–58, July 2009.

[28] O. Ogorodnikova, "Methodology of safety for a human robot interaction designing stage," in *2008 Conference on Human System Interactions*, pp. 452–457, IEEE, 2008.

[29] M. Vasic and A. Billard, "Safety issues in human-robot interactions," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 197–204, IEEE, 2013.

[30] D. Park, H. Kim, and C. C. Kemp, "Multimodal anomaly detection for assistive robots," under review.

[31] P. Ahrendt, "The multivariate gaussian probability distribution," tech. rep., 2005.

[32] G. F. Kuhn, "Model for the interaural time differences in the azimuthal plane," *The Journal of the Acoustical Society of America*, vol. 62, no. 1, pp. 157–167, 1977.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[35] F. Chollet, "Keras." `https://github.com/fchollet/keras`, 2015.

[36] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[37] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The International Journal of Robotics Research*, p. 0278364912471865, 2013.

[39] T. Bhattacharjee, A. Jain, S. Vaish, M. D. Killpack, and C. C. Kemp, "Tactile sensing over articulated joints with stretchable sensors," in *World Haptics Conference (WHC), 2013*, pp. 103–108, IEEE, 2013.

[40] A. Weiss, R. Bernhaupt, M. Lankes, and M. Tscheligi, "The usus evaluation framework for human-robot interaction," in *AISB2009: proceedings of the symposium on new frontiers in human-robot interaction*, vol. 4, pp. 11–26, 2009.