Autonomous Robots (2019) 43:611–629
https://doi.org/10.1007/s10514-018-9733-6

Multimodal anomaly detection for assistive robots

Daehyung Park[1] · Hokeun Kim[1] · Charles C. Kemp[1]

Received: 25 December 2016 / Accepted: 2 April 2018 / Published online: 13 April 2018

Abstract
Detecting when something unusual has happened could help assistive robots operate more safely and effectively around people. However, the variability associated with people and objects in human environments can make anomaly detection difficult. We previously introduced an algorithm that uses a hidden Markov model (HMM) with a log-likelihood detection threshold that varies based on *execution progress*. We now present an improved version of our previous algorithm (HMM-D) and introduce a new algorithm based on Gaussian process regression (HMM-GP). We also present a new and more thorough evaluation of 8 anomaly detection algorithms with force, sound, and kinematic signals collected from a robot closing microwave doors, latching a toolbox, scooping yogurt, and feeding yogurt to able-bodied participants. Overall, HMM-GP had the highest performance in terms of area under the curve for these real-world tasks, and multiple modalities improved performance with some anomalies being better detected with particular modalities. With synthetic anomalies, HMM-D exhibited shorter detection delays and outperformed HMM-GP with high-magnitude anomalies. In general, higher-magnitude synthetic anomalies tended to be detected more rapidly.

Keywords Multimodality · Anomaly detection · Assistive manipulation · Execution monitoring

1 Introduction

Activities of daily living (ADLs), such as feeding and dressing, are important for quality of life (Wiener et al. 1990). Robotic assistance could help people with disabilities perform ADLs on their own, such as robot-assisted shaving (Hawkins et al. 2014), robot-assisted wiping (Wakita et al. 2012; King et al. 2010), and robot-assisted feeding (Schrer et al. 2015; Copilusi et al. 2015a; Takahashi and Suzukawa 2006; Park et al. 2016b). Similarly, robotic assistance with instrumental activities of daily living (IADLs) like household chores (Silvrio et al. 2015; Leidner et al. 2016) could be valuable. When operating near humans, it is particularly important that robots safely handle anomalous task executions. A robot can monitor the execution of a task using a separate system that runs in parallel: a type of execution monitoring system (Pettersson 2005). By monitoring multimodal sensory signals, an execution monitor could perform a variety of roles, such as detecting success, determining when to switch behaviors, and otherwise exhibiting more common sense (Fig. 1).

In this paper, we focus on using an execution monitor to detect anomalous executions. We consider an execution to be anomalous when it is significantly different from typical successful executions, which does not strictly imply that an anomalous execution is a failure. For example, an execution might be anomalous due to high uncertainty about the outcome, but still result in successful completion of the task if allowed to proceed. Due to variability during manipulation in human environments (Kemp et al. 2007), anomaly detection is challenging. An ideal execution monitor should be capable of detecting anomalies online, alerting the robot shortly after the onset of an anomaly, working for long-duration behaviors, detecting subtle anomalies, ignoring irrelevant task variation, and handling multimodal sensory signals. In pursuit of these goals, we have developed a data-driven anomaly detection framework that uses multimodal monitoring (see Fig. 2).

The principles underlying our approach are twofold. The first is that multimodal sensory signals improve detection since each anomaly is relevant to a distinct set of modali-

✉ Daehyung Park
deric.park@gatech.edu

Hokeun Kim
hkim751@gatech.edu

Charles C. Kemp
charlie.kemp@bme.gatech.edu

[1] Healthcare Robotics Lab, Georgia Institute of Technology, Atlanta, GA, USA

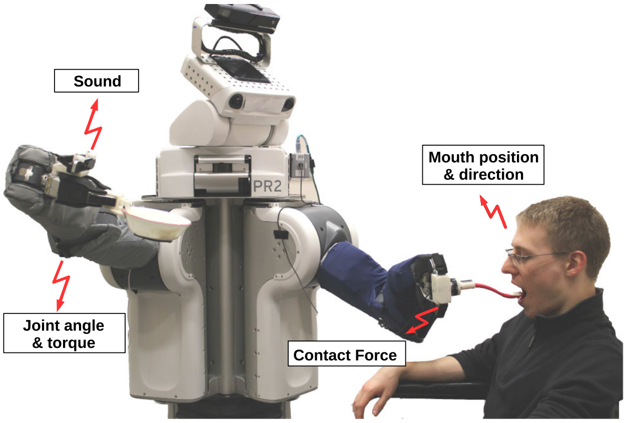


Fig. 1 Multimodal monitoring in a robotic feeding task. A PR2 robot determines anomalous executions collecting haptic, auditory, kinematic, and visual sensory signals

ties, and cross-modal anomalies may not be evident when monitoring modalities independently. To detect anomalous patterns in signals, researchers have used Markovian modeling (Chandola et al. 2009) and computed likelihoods, state paths, or state distributions to classify anomalies over time (Warrender et al. 1999; Perduca and Nuel 2013). As with our previous work (Park et al. 2016a), our algorithms use multivariate Gaussian hidden Markov models (HMMs) to jointly model multimodal time-series signals.

Our second underlying principle is that changeable decision boundaries are beneficial to handling task variability, to detecting anomalies quickly, and to reducing the number of false alarms. In our work, all signals come from executions of a specific robot behavior (e.g., linear motion) applied to a particular task (e.g., closing a door or feeding someone). As with our previous work (Park et al. 2016a), our algorithms use a detection threshold that depends on the *execution progress*. Specifically, our algorithms estimate the mean and standard deviation of the HMM's log-likelihood given the current execution progress. If the HMM's log-likelihood is unusually low given the execution progress, our algorithms report an anomaly.

In this paper, we improve the algorithm, HMM-D, from our previous work (Park et al. 2016a), which uses clusters to estimate the mean and standard deviation of the HMM's log-likelihood given the execution progress. Cluster membership for HMM-D is based on temporal similarity. We also evaluate a new algorithm, HMM-KNN, that uses clusters based on execution progress similarity. In addition, we introduce a new algorithm, HMM-GP, that uses Gaussian process regression to estimate the mean and standard deviation of the HMM's log-likelihood given the execution progress.

To evaluate our methods, we collected multimodal signals such as haptic, auditory, kinematic, and vision signals while a PR2 robot performed pushing and feeding tasks for IADLs and ADLs, respectively (Sakaguchi et al. 2009; Topping 2002; Park et al. 2016b). We evaluated our method with a PR2 robot, which repeatedly closed the door of a microwave oven or the lid of a toolbox. We also ran a study in which a PR2 robot fed yogurt to able-bodied participants, scooping yogurt from a bowl and bringing it to their mouths (see Fig. 1). In evaluations, we compared several performance scores with simulated and experimental anomalous data. Our multimodal monitoring outperformed other baseline monitoring methods, obtaining a higher area under curve (AUC) from receiver operating characteristic (ROC) curves and shorter detection delays. We also show that multimodalities substantially improved detection performance and detected a broad range of anomalies.

2 Related work

2.1 Assistive robot

Researchers have introduced robotic devices to augment user functionality such as mobility (Papageorgiou et al. 2014;

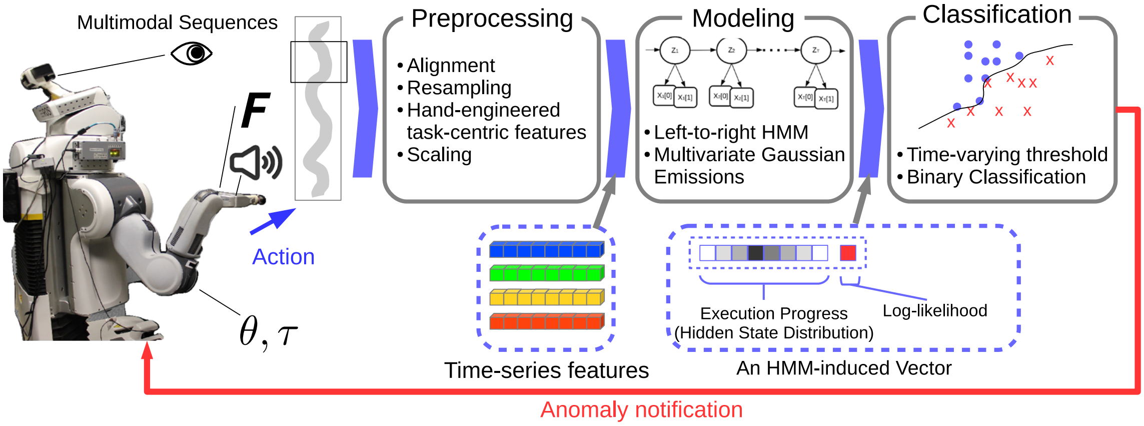


Fig. 2 The framework of our proposed execution monitoring system for anomaly detection

Argall 2016), shaving (Chen et al. 2013), or picking-and-placing (Graf et al. 2009; Nguyen et al. 2008; Ciocarlie et al. 2012). We are particularly interested in robotic feeding assistance that has been explored in (Topping 2002; Song and Kim 2012; Copilusi et al. 2015b). Many commercialized feeding robots are currently available: Bestic arm (Jiménez Villarreal and Ljungblad 2011), Meal Buddy (Patterson Medical 2017), Obi (Eclipse Automation 2016), and Mealtime partner (Mealtime Partners 2017). However, the specialized design of these robots restricts their ability to generalize to other tasks. In this paper, we use a general-purpose mobile manipulator with autonomous motions for various assistive robotic tasks.

In many instances, physical interaction is unavoidable between assistive robots and users. To prevent potential hazards, many studies have used simple safety constraints. Hawkins et al. (2014) detected an anomaly in shaving assistance when contact force was over a threshold. Kim et al. (2014) detected a grasping failure when contact forces changed less than a threshold in sliding time windows. Yamazaki et al. (2009) detected failure of IADL tasks when no significant visual change occurred. These methods are relatively straightforward to implement, but are limited in the types of anomalies they can detect.

2.2 Execution monitoring

State Estimation Execution monitoring has been well-studied in robotics for the detection and the isolation of anomalous executions (Pettersson 2005; Bouguerra et al. 2008). Monitoring methods usually estimate the state of a system from observations and determine anomalies with respect to that state. For example, Jain and Kemp (2013) estimated a task-centric state (i.e., opening angle) from observed kinematics during a door-opening task and detected blocked or locked doors. Haidu et al. (2015) split a trajectory into a number of bins as states and determined anomaly decision boundaries per bin. Likewise, Kappler et al. (2015) split multimodal signals, such as force, audio, and kinematics data, into discretized time states and trained a linear support vector machine as a failure classifier per bin. Simanek et al. (2015) estimated the state of a mobile robot from multimodal data using an extended Kalman filter and rejected anomalous data using statistical tests. The mentioned methods assumed that a state could only be determined by current observations. Unlike the directly observable states they used, we represent the current progress of a task execution as the distribution over hidden states of an HMM.

Anomaly Detection Anomaly detection, the main role of execution monitoring, has been widely explored in various domains, including credit-card fraud, cyber intrusion, and novelty detection (Markou and Singh 2003; Chandola et al. 2009; Dua and Du 2011; Pimentel et al. 2014). A number of robotics researchers have also applied it to robotics applications: mechanical-failure detection (Bittencourt et al. 2014; Brambilla et al. 2008), sensor-fault detection (Blank et al. 2011; Suarez et al. 2016), and environmental anomaly detection (Lepora et al. 2010; Dames et al. 2016). These studies have also been applied to the detection of the anomalous statuses of highly complex systems such as humanoid robot falls (Suetani et al. 2011; Marcolino and Wang 2013). As an extension, similar to our work, researchers are attempting to detect task-dependent failures: blocked door (Jain and Kemp 2013) and bin-picking failure (Rodriguez et al. 2011). For example, Sukhoy et al. (2012) detected manipulation failures in a magnetic card sliding task using multiple torque outputs. Rather than using static decision boundaries, we use dynamic decision boundaries that depend on the currently perceived state of the task to improve the detection performance.

2.3 Multimodal time-series modeling

Multimodality Researchers have taken two distinct approaches to modeling time-series sensory signals. In the first, a model learns its internal structure from each sensory modality independently. Rodriguez et al. (2010) classified the failure of a robotic assembly by extracting a unimodal signal, the magnitude of force output. Ando et al. (2011) determined the anomalous behaviors of a mobile robot using k-nearest neighbors, which measures the difference between modalities independently. Pastor et al. (2011) used multimodal sensory signals to predict failures while making a robot flip a box using chopsticks. Their method independently modeled each signal using dynamic movement primitives and predicted a failure when five or more signals failed independent z-tests for three consecutive time steps with respect to recorded signals from successful attempts. Chu et al. (2013) trained multiple HMMs with respect to each modality and input a set of log-probabilities from the HMMs to a linear SVM for haptic adjective classification. In the second, a model learns multivariate representations from multimodal sensory signals. Clifton et al. (2011) represented the multimodal distribution of data using multivariate Gaussian mixture models (GMM) and detected anomalies using extreme value theory. In our previous work (Park et al. 2016a), we modeled multimodal sensing signals using the multivariate Gaussian emission probabilities of an HMM. This approach, although computationally expensive, is capable of checking anomalies in which two or more signals jointly change.

Deep Learning Researchers have applied neural networks to anomaly detection. A representative approach has been the use of reconstruction-based methods that detect high reconstruction error as an anomaly using multilayer perceptrons

(MLPs) (Lee et al. 2002), long term short term memory networks (LSTMs) (Malhotra et al. 2015), or other recurrent neural networks (RNNs) (Williams et al. 2002). Alternatively, in a brief workshop paper, Sölch et al. (2016) described a generative model for anomaly detection that uses a variational autoencoder with recurrent neural networks. For their evaluation, they first trained the neural network on the joint angles of a 7 degree-of-freedom robot arm performing a pick and place task. They then tested detection of anomalous collisions with the arm using this neural network with various thresholds. Their method shares similarities with our approach, although they did not present results for multimodal anomaly detection.

Time-series Data Our research is closely related to time-series data classification. Fujii et al. (2016) determined material defects in hammering tests by continuously matching time-frequency audio images with template patterns. Niekum et al. (2014) detected motion change points using the evidence probabilities of articulation models given a segment of time-series data. Similar to our work, before classification, a number of researchers modeled time-series data using dynamics models. Fagogenis et al. (2016) modeled time-series data using locally weighted projection regression to check the thruster failures of autonomous underwater vehicles. Morris and Trivedi (2008) used an HMM to model object paths for abnormal behaviors. Mendoza et al. (2014) also used an HMM to model multiple signals such as velocity, acceleration, and jerk signals. However, different from our approach, these studies assumed conditional independence between signals even though the signals may have been correlated. We instead use full covariance between signals in HMMs.

2.4 Thresholding

Researchers have determined a fault when a raw sensory signal exceeds a fixed threshold (Angelov et al. 2006). However, due to the range of the signals in various operations, a threshold had to be large enough to minimize frequent false alarms. Narrowing down the range of operations, Serdio et al. (2014) introduced a time-varying threshold, allowing a tight decision boundary around low variance area for rolling mills. However, time variations, such as delay or speed variation, easily break down the detection system.

Instead, researchers have modeled time-series signals and determined anomalies or new events when the likelihood of current observations is lower than a fixed threshold (Lühr et al. 2004; Khan et al. 2012). For example, Vaswani et al. (2005) detected abnormal activities when the expected negative log-likelihood was lower than a fixed threshold. Di Lello et al. (2013) detected anomalous force signals using a fixed threshold, and Ocak and Loparo (2005) classified anomalies when either likelihood exceeded a fixed threshold or a change in the likelihood between time steps exceeded another fixed threshold. However, these thresholding methods are not applicable if the likelihood does not drop significantly or quickly. Setting a global threshold may reject a number of non-anomalous executions, though it may also fail to detect all anomalies (Markou and Singh 2003).

To address these concerns, Rodriguez et al. (2011) set cut-off probability thresholds over discretized time slices. Yeung and Ding (2003) used a varying likelihood threshold based on the probabilistic distribution of discrete observations. The use of a time-varying likelihood threshold is similar to our work, but we do not directly use observations. Instead, we vary the threshold with respect to the hidden-state distribution, execution progress, estimated by an HMM.

3 Task-centric multimodal sensory signals

Robots are able to access many sensory devices that may be informative during a task execution. We have observed haptic, auditory, visual, or kinematic sensory signals that are particularly relevant to manipulation tasks. Figure 3 shows an example of how two task-relevant signals, force and sound, change as a PR2 robot closes a microwave door. Depending on the state of the spring latch, associated forces and sounds vary. We can observe similar patterns of signals from pushing tasks with various objects, in which the robot repeatedly pushes with its end effector to close or turn on an object in a fixed amount of time. Figure 4 provides a visualization of force and sound changes recorded for 30 non-anomalous executions with three objects. At any point, an anomaly can result in changes in a modality or multiple modalities. In particular, in Fig. 4a, a small force anomaly may occur in the high variance area. Its detection will be non-trivial given only the force modality. We can complement the lack of information using other modalities such as the sound that co-occurs with the force.

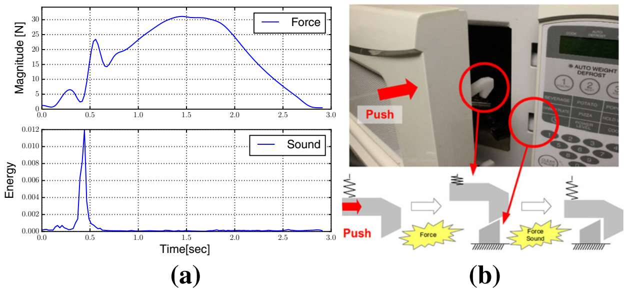


Fig. 3 Changes in the force and sound magnitudes during the execution of a microwave door closing task. Note that the force magnitude continues to increase even after latching since the operator, a PR2 robot, was programmed to push the door for a fixed duration of time. **a** Distribution of force and sound, **b** latching mechanism

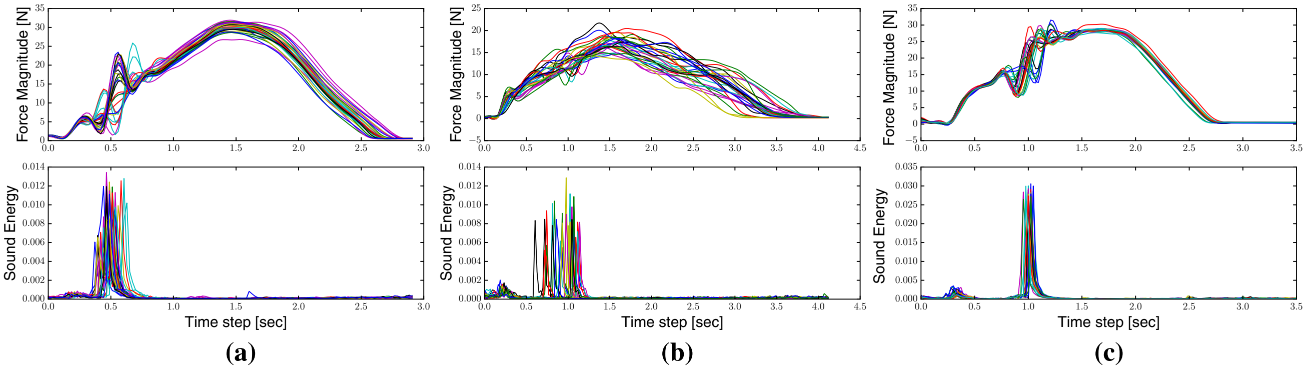


Fig. 4 Visualization of the force and sound sequences recorded in three representative manipulation tasks: closing microwave doors and latching a toolbox. **a** Microwave (white), **b** toolbox **c** Microwave (black)

To model and monitor a task, we extract hand-engineered task-centric features from multimodal sensory signals. Other researchers have used automatic feature extraction methods for event detection or recognition using principal component analysis (PCA) (Hoffmann 2007) or autoencoders (Yu and Seltzer 2011; Gehring et al. 2013). However, anomalies do not necessarily come from the dominant components of observations. In other words, after the PCA, anomalies may not be detectable if their relevant signals were stationary in the training data. Instead, we use hand-engineered features based on our domain knowledge (see Sect. 6).

Our system represents sensory signals such as position and orientation with respect to a task, particularly an object. The task-centric representations are beneficial as they are pose-invariant from initial conditions of a PR2 and a target object. Inherently pose-invariant features such as sound and force are taken from raw input. We then extract hand-engineered features from raw sensory inputs. Note that after collecting raw sensory signals, we resample them to obtain the same sequence length since modalities have an imbalanced amount of information with distinct sensing frequencies. After extracting features $\mathbf{X}$, we scale each feature $\mathbf{x} \in \mathbf{X}$ individually to a given range, i.e., between 0 and 1, $\mathbf{x} = (\mathbf{x} - \mathbf{x}_{\min})/(\mathbf{x}_{\max} - \mathbf{x}_{\min})$.

4 Generative modeling

We formulate the problem of anomaly detection as the estimation of joint distribution $P(\mathbf{X}|\mathbf{X}_{\text{train}})$ given a set of training data $\mathbf{X}_{\text{train}}$ from normal activities, where $\mathbf{X}$ represents a sequence of multimodal observations (or task-centric features). When the estimated probability is lower than a threshold (see Sect. 5), our execution monitoring system determines $\mathbf{X}$ to be *anomalous*. In this section, we describe how to model the training data and list useful output for the estimation of a joint distribution.

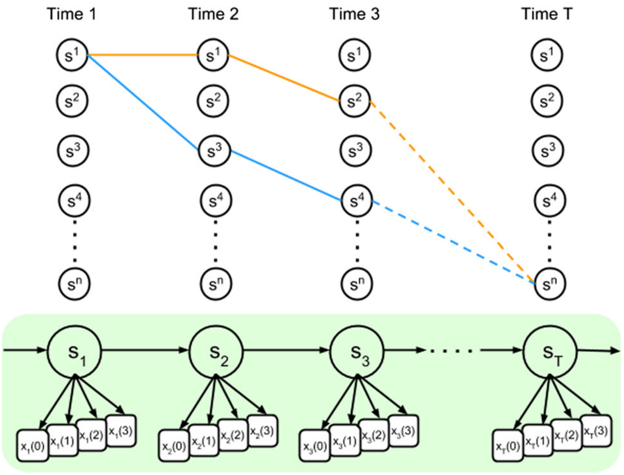


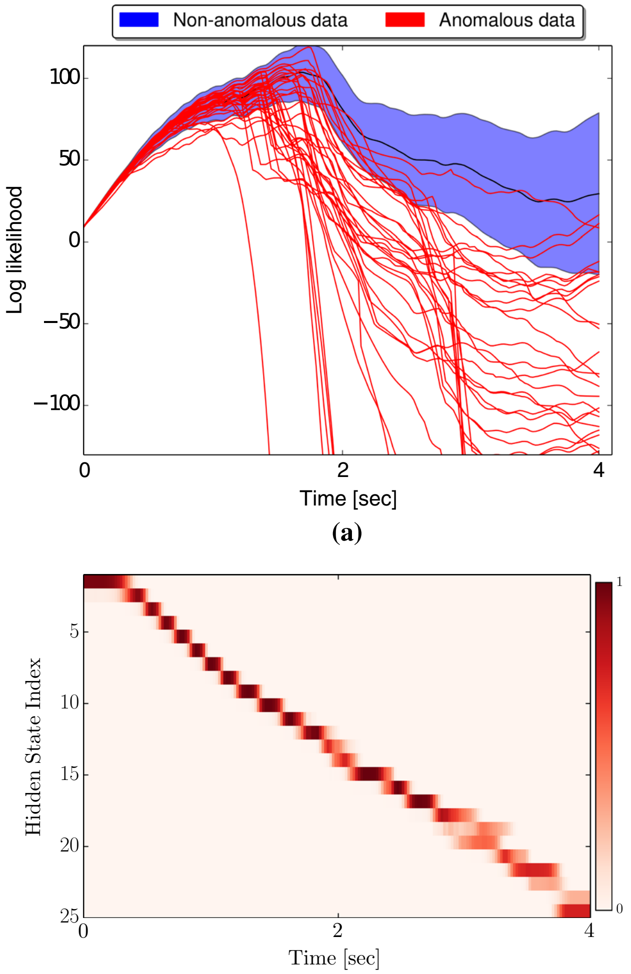
Fig. 5 Architecture of left–right hidden Markov models with multivariate Gaussian emissions ($m = 4$)

4.1 Hidden Markov models (HMMs)

We model extracted features using a left-to-right HMM with multivariate Gaussian emissions. Figure 5 depicts the architecture of the HMM and two examples of hidden-state paths (blue and red lines) associated with a time series of multidimensional observations. Let random variable $\mathbf{x}_i$ be an m-dimensional observation (i.e., feature) vector at time step i. We represent a sequence of observations as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i\}$. Let random variable $\mathbf{z}_i^j$ be the jth hidden state out of n hidden states at time step i. The left-to-right structure forces state indices to remain constant or increase over time. To represent multi-dimensional output, we use a multivariate Gaussian distribution with a full covariance matrix so that we can represent correlations among modalities.

Collecting data across the entire space of anomalous executions is not feasible, so we use only negatively labeled *non-anomalous* data as training data. Assuming that a robot performs stereotyped task-specific behaviors, the training data $\mathbf{X}_{\text{train}}$ contains consistent patterns so that we can represent $\mathbf{X}_{\text{train}}$ as a set of model parameters λ_{MAP} through maximum a-posteriori (MAP) estimation. For λ_{MAP}, we use the Baum-Welch algorithm. We can approximate the estimation of the joint distribution as $P(\mathbf{X}|\lambda_{\text{MAP}})$. For convenience, we omit the subscript "MAP" when using parameter set λ.

Before training an HMM, we initialize the initial state distribution π and the transition probability matrix $\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ to use the left-to-right structure. We set the first element of the n-dimensional vector π to 1.0 and all other elements to 0.0 in order to start the HMM in the first state. We also set $\mathbf{A}$ to be an upper triangular matrix with linearly decreasing transition probabilities from 0.4 to 0.0 for gradual left-to-right state transitions. In this work, we use the General Hidden Markov Model library (GHMM) (Schliep et al. 2004).



(a)

(b)

Fig. 6 An example of log-likelihood and execution progress distributions (i.e., hidden-state distributions) when a PR2 robot closes the door of a microwave oven (white). **a** The blue-shaded region and red curves show the standard deviation of log-likelihoods from 35 non-anomalous executions and the unexpected drop of log-likelihoods from 18 anomalous executions, respectively. **b** This graph shows averaged changes of execution progress vectors from the non-anomalous executions (Color figure online)

4.2 An HMM-induced vector

While performing a task, an HMM can generate informative output such as likelihood, posterior probability distribution, or transition probabilities (Chu et al. 2013; Ketabdar et al. 2006). We refer to a list of the outputs as an *HMM-induced vector*. The most commonly used output for classification is log-likelihood $l = \log P(\mathbf{X}|\lambda)$, which we can calculate from the sum of joint distributions, $P(\mathbf{X}|\lambda) = \sum_{\mathbf{Z}} P(\mathbf{X}, \mathbf{Z}|\lambda)$ (Bishop 2006), where $\mathbf{Z}$ is a state path over hidden state space $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ and T is the last time step in $\mathbf{X}$. Another output is the posterior probability distribution $P(\mathbf{z}_t|\mathbf{X}, \lambda)$, which we use as execution progress that represents the progress of a task-specific behavior at time t.

During non-anomalous executions, the likelihood tends to vary in consistent ways (see the upper graph in Fig. 6) so we can model this variation with respect to the progress. The property of the left-to-right model is that hidden states must be in non-decreasing order such as $\mathbf{Z} = \{\mathbf{z}_1^1, \mathbf{z}_2^1, \mathbf{z}_3^2, \mathbf{z}_4^3, \mathbf{z}_5^4, \ldots, \mathbf{z}_T^n\}$.[1] If the true states were known, their indices could represent the progress. Compared to the direct use of time, the representation would have the advantage of handling variability in the timing of a behavior execution, but the true state path is hidden from the observer. One approach can represent the progress as the state indices with the maximum likelihood at any given moment, though the indices would neglect uncertainty. Instead, we represent the progress, $\gamma(t)$, as a probability mass function over hidden states (the hidden-state distribution) at time t, $\gamma(t) = P(\mathbf{z}_t|\mathbf{X}, \lambda)$. The lower graph in Fig. 6 shows that the execution progress averaged across all non-anomalous trials can change in an intuitive way with respect to time with the index of the most likely state progressively increasing. Note that HMMs will be able to generate a new dataset close to $\mathbf{X}_{\text{train}}$ given the execution progress but we do not investigate the generative properties in this paper. We compute the n-dimensional vector $\gamma(t)$ with the forward and backward procedures of the EM algorithm (Rabiner 1989),

$$\gamma(t) = \frac{P(\mathbf{X}(1:t), \mathbf{z}_t \mid \lambda) \cdot P(\mathbf{X}(t+1:T) \mid \mathbf{z}_t, \lambda)}{P(\mathbf{X} \mid \lambda)}, \tag{1}$$

where T is the last time sample of $\mathbf{X}$.

Figure 7 illustrates an example of execution progress and likelihood changes during a non-anomalous and an anomalous trial of a closing task. The upper two subgraphs show force and sound observations (blue curves) over time. Each

[1] The state path of an HMM always starts from the first hidden state, $\mathbf{z}^1$, setting $\pi = \{1, 0, \ldots, 0\}$.

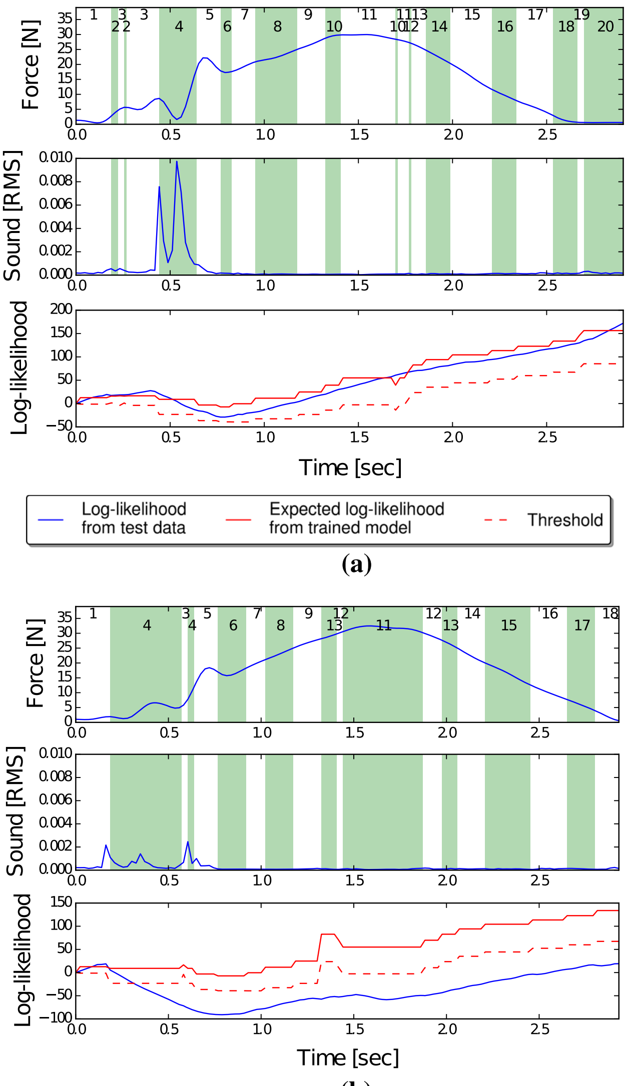


Fig. 7 Comparison of non-anomalous and anomalous observations in the door-closing task for a microwave (white). The upper two graphs for **a**, **b** show the force and sound observations over time. Each white or green band denotes a period of time over which the most likely hidden state remains constant. The small black number in each band is the index for the most likely hidden state over the duration of the band. These indices need not increase monotonically since we computed them in an online fashion using only previous observations at each time step. The bottom graphs in **a**, **b** illustrate the expected log-likelihood based on the execution progress (solid red curve), the log-likelihood resulting from the ongoing trial (solid blue curve), and the threshold based on execution progress (related to the dashed red curve). For this comparison, we set $c = 2.0$ and blocked the door using a rubber pad for the anomalous operation (Color figure online)

white or green band represents a period of time over which the most likely hidden state remains constant. Each small black number indicates the index for the most likely hidden state over the duration of a band. These indices need not increase monotonically since we computed them in an online fashion using only prior observations at each time step. The blue curves of the bottom subgraphs show changes in the log-likelihoods, which we will discuss in Sect. 7.

5 Anomaly classification

Our framework uses a binary, or one-class, classification method that learns from only non-anomalous execution data. It detects an anomaly when the log-likelihood of a sequence of observations X is lower than a varying threshold at some point in time during the execution. Otherwise, it considers the execution to be non-anomalous. We represent a threshold as $\hat{\mu}(\gamma) - c\hat{\sigma}(\gamma)$, where $\hat{\mu}$, $\hat{\sigma}$, and c are the expected log-likelihood, its standard deviation (or confidence interval), and a constant that adjusts the sensitivity of the classifier, respectively. For notational clarity, we omit t from $\gamma(t)$. In this section, we introduce two methods of estimating $\hat{\mu}$ and $\hat{\sigma}$ using clustering and regression methods.

5.1 Clustering-based threshold (HMM-D)

We first introduce a clustering-based method that provides a dynamically changing threshold. We call this method HMM-D that is an improved version from (Park et al. 2016a). HMM-D clusters and parameterizes the HMM-induced vectors associated with only non-anomalous executions into K soft clusters. In the training step, we use Gaussian radial basis functions (RBFs) to produce the clusters. Assuming we have a similar phase of executions, we set evenly-distributed clusters over time that weight the membership of HMM-induced vectors based on RBFs. We then parameterize each cluster defined by a 3-tuple consisting of a weighted average of execution progress vectors, a weighted mean of log-likelihood vectors, a weighted standard deviation of log-likelihood vectors. The RBFs provide the weight. In the testing step, these clusters are then used to map execution progress vectors to estimates of the mean and standard deviation of the log-likelihood.

Figure 8 illustrates the distribution of RBFs, each of which weights execution progress vectors differently. For example, the kth cluster weights a vector at time t with its associated RBF $\phi(t, w_k) = e^{-\epsilon(t-w_k)^2}$, where $k \in \{1, \ldots, K\}$, w_k is the center of the kth RBF, and ϵ is a constant. Note that we use fixed length M of the time series in $\mathbf{X}_{\text{train}}$. We compute a weighted average of *execution progresses* $\hat{\gamma}_k$ at the kth cluster using

$$\hat{\gamma}_k = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{\eta_k} \sum_{t=1}^{T} \gamma^{(i)}(t) \phi(t, w_k) \right], \tag{2}$$

where N is the number of non-anomalous time series in $\mathbf{X}_{\text{train}}$, η_k is a normalization factor, $\eta_k = \sum_{t=1}^{T} \phi(t, w_k)$, and $\gamma^{(i)}(t)$

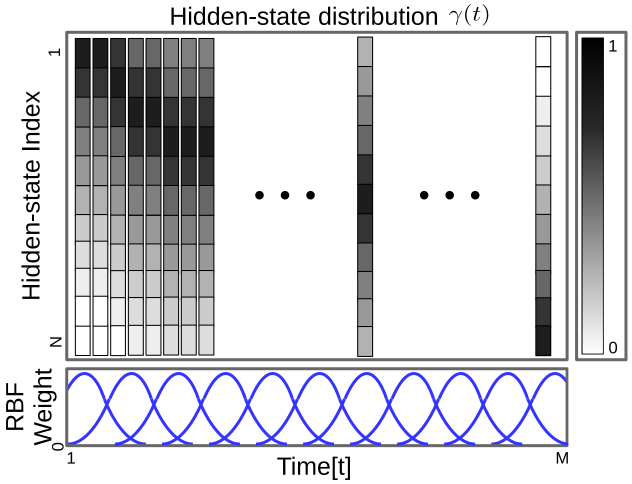


Fig. 8 Illustration of the K clusters of execution progress vectors (i.e., hidden-state distributions). Each cluster has an associated RBF used to weight execution progress vectors $\gamma(t)$, and their associated log-likelihoods $L(t)$ based on when they occurred in time. These weights are used to compute $\hat{\gamma}_k$, $\hat{\mu}_k$, and $\hat{\sigma}_k$ for cluster k

denotes the execution progress at time t for the ith time series ($i \in \{1, \ldots, N\}$).

Likewise, we also compute weighted mean $\hat{\mu}_k$ and variance $\hat{\sigma}_k$ of log-likelihoods at the kth cluster,

$$\hat{\mu}_k(L) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{\eta_k}\sum_{t=1}^{T} L^{(i)}(t)\phi(t, w_k)\right], \tag{3}$$

$$\hat{\mu}_k(L \cdot L) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{\eta_k}\sum_{t=1}^{T} L^{(i)}(t)^2\phi(t, w_k)\right], \tag{4}$$

$$\hat{\sigma}_k(L) = \sqrt{\hat{\mu}_k(L \cdot L) - (\hat{\mu}_k(L))^2}, \tag{5}$$

where L denotes a set of log-likelihoods along non-anomalous time series,

$$L = \{L^{(1)}(0), \ldots, L^{(1)}(T), \ldots, L^{(N)}(0), \ldots, L^{(N)}(T)\}, \tag{6}$$

and $L^{(i)}(t)$ denotes a log-likelihood at time t for the ith time series. We parameterize and store the K clusters as

$$\{(\hat{\gamma}_1, \hat{\mu}_1, \hat{\sigma}_1), (\hat{\gamma}_2, \hat{\mu}_2, \hat{\sigma}_2), \ldots, (\hat{\gamma}_K, \hat{\mu}_K, \hat{\sigma}_K)\}, \tag{7}$$

where we omit L from $\hat{\mu}_k(L)$ and $\hat{\sigma}_k(L)$ for notional clarity.

At run time, we compute $\gamma(t)$ and $\log P(\mathbf{X}_t|\lambda)$ at time step t from the incoming signals. The detector then finds the index of the closest cluster, k^*, by comparing the difference between γ and each of the K clusters using symmetric Kullback-Leibler divergence (SKL-divergence),

$$k^* = \underset{1,\ldots,K}{\arg\min}\ D_{\mathrm{SKL}}(\gamma(t)||\gamma_k), \tag{8}$$

$$= \underset{1,\ldots,K}{\arg\min}\ \min\left(D_{\mathrm{KL}}(\gamma(t)||\gamma_k), D_{\mathrm{KL}}(\gamma_k||\gamma(t))\right) \tag{9}$$

where $D_{\mathrm{KL}}(P||Q)$ is a measure of the information lost when Q is used to approximate P. The system determines an anomaly with following comparison:

$$\begin{cases} \text{anomaly} & \text{if } \log P(\mathbf{X}_t \mid \lambda) < \hat{\mu}_{k^*} - c\hat{\sigma}_{k^*} \\ \neg\text{anomaly}, & \text{otherwise} \end{cases} \tag{10}$$

where c is a real-valued constant to determine a sensitivity. Increasing c tends to result in fewer reported anomalies and an accompanying lower false positive rate and lower true positive rate. Decreasing c tends to result in more reported anomalies and an accompanying higher false positive rate and higher true positive rate.

In order to use HMM-D with training data from successful executions with large time variations, users would most likely need to align the data in time using dynamic time warping or another method. However, after training, anomaly detection would not require time-warped signals since the system only uses execution progress to estimate the mean and standard deviation of the log-likelihood. While we do not explicitly test tasks that involved large timing variations, our tasks have timing variations as illustrated in Fig. 4. During the feeding task in Sect. 7, human actions also resulted in timing variations. In addition, to help clarify the role of the RBFs, we also introduce a state-based clustering method using k-nearest neighbors (k-NN) without using the RBFs. We call this method as HMM-KNN which will be discussed in Sect. 7.

5.2 Regression-based threshold (HMM-GP)

We introduce an alternative detection method that uses regression to estimate the mean and standard deviation of the log-likelihood with respect to the execution progress. In the previous algorithms, the discontinuities between clusters may lower detection performance. To address this potential issue, we can apply any parametric or non-parametric regression method that provides mean and variance of the likelihood for our detector. In this paper, we use a Gaussian process regressor, referred to as GP, since it is also well supported by the community with open-source libraries and a number of variants (Snelson and Ghahramani 2006; Rasmussen et al. 2003). We serially connect an HMM and a GP regression method that is able to output smooth detection thresholds with a comparably smaller amount of training data. It maps input and output pairs to predict an output given a new input with confidence intervals (Rasmussen 2004).

Let D be the length of an execution progress vector. Given training input and output pairs ($\boldsymbol{\gamma} \in \mathbb{R}^{NT \times D}$, $L \in \mathbb{R}^{NT}$), Gaussian process can model the predictive distribution of log-likelihood $l_* \in \mathbb{R}$ at an execution progress point $\gamma_* \in \mathbb{R}^D$ as

$$P(l_* \mid \gamma_*, L, \boldsymbol{\gamma}) = \mathcal{N}(\mu_*, \Sigma_*), \tag{11}$$

where μ_* and Σ_* are posterior mean and covariance estimates, respectively. In this work, we use a squared exponential function,

$$k(\gamma, \gamma') = \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(\gamma(d) - \gamma'(d))^2}{l_d}\right) \tag{12}$$

to represent the covariance of $\mathbf{x}$ where l_d is an individual length scale hyperparameter for each input dimension d. We use a GP to find l_* that maximizes the marginal likelihood using

$$\mu_* = \mathbf{k}_*^T \mathbf{K}^{-1} L, \tag{13}$$

$$\Sigma_* = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*, \tag{14}$$

where $\mathbf{K} = k(\boldsymbol{\gamma}, \boldsymbol{\gamma})$, $\mathbf{k}_* = k(\boldsymbol{\gamma}, \gamma_*)$, and $k_{**} = k(\gamma_*, \gamma_*)$. Through this process, we can find the output l_* with variance Σ_* given an observed execution progress.

To classify anomalies, HMM-GP also uses a statistical decision similar to the detection criteria described in Eq. (10):

$$\begin{cases} \text{anomaly} & \text{if } \log P(\mathbf{X}_t \mid \lambda) < \mu_* - c\sqrt{\Sigma_*} \\ \neg\text{anomaly}, & \text{otherwise} \end{cases} \tag{15}$$

where c is a constant that adjusts the sensitivity of the detector. Note that the covariance estimate Σ_* may have undergone undesired changes for each point by an ill-posed covariance matrix. To regularize it, we add constants, called nugget values (Cressie 1993), into the diagonal elements of the training-data covariance matrix to get smooth changes of the output distribution.

In terms of the amount of data, GP's non-parametric regression can be computationally expensive due to a large covariance matrix. In this paper, we randomly subsampled training data to acquire a maximum of 1000 samples to avoid the issue, and it performed well in Sect. 7. Other methods are also available to handle this issue, such as sparse Gaussian process (SPGP) (Snelson and Ghahramani 2006).

6 Experimetal setup

We evaluated our approach with pushing and robot-assisted feeding tasks selected from IADLs and ADLs. The pushing task includes behaviors such as closing doors and latching a toolbox. The feeding task includes scooping yogurt from a bowl and delivering spoonfuls of yogurt to the mouths of able-bodied participants. We conducted this research with approval from the Georgia Tech Institutional Review Board (IRB), and obtained informed consent from all participants.

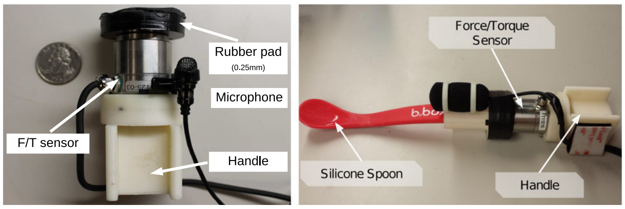


Fig. 9 Each instrumented tool for pushing and scooping tasks has a force-torque sensor and a microphone mounted on a 3D-printed handle, designed to be held by the PR2 gripper. Left: A pushing tool with a rubber-padded plastic circle. Right: A scooping tool with a flexible silicone spoon

6.1 Instrumentation for multimodal sensing

The PR2 robot that we used is a 32-DOF mobile manipulator with two 7-DOF back-drivable arms and powered grippers controlled by a 1 kHz low-level PID controller. Its maximum payload and grip force are listed as 1.8 kg and 80 N, respectively. We made the PR2 hold a tool and mounted sensors on it.

For the pushing and scooping behaviors, the PR2 held an instrumented tool with a 3D-printed handle designed for its grippers (see Fig. 9). The tool included a force/torque sensor (ATI Nano25) and a unidirectional microphone for monitoring haptic and auditory signals. During manipulation, our monitoring system recorded 6-axis force/torque measurements at 1 kHz, and simultaneously recorded sound input from the microphone at 44.1 kHz. We also used two additional cross-modal features, relative position and relative orientation between the tool and a target object. The system determined its tool pose using joint encoder values and forward kinematics. It estimated a target pose using a Microsoft Kinect V2 and an ARTag (Fiala 2005) attached to the door of the microwave oven at 10 kHz.

For the feeding behavior, the PR2 also held a tool with a force/torque sensor. To measure a person's mouth pose as a target, we used an RGB-D camera (Intel SR300) mounted on the wrist of one arm (see Fig. 10).

6.2 Feature selection and preprocessing

We extracted hand-engineered, task-centric features from multimodal sensory signals to train a trans-invariant detector since the coordination between a robot and a target varies

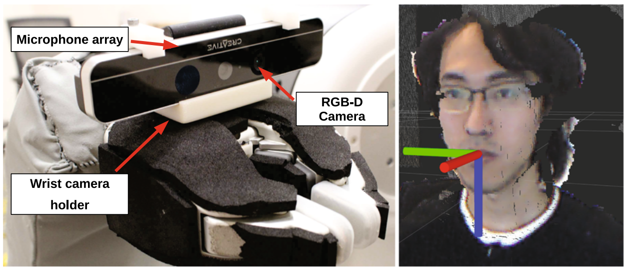


Fig. 10 A wrist-mounted sensing tool for the feeding task. Left: An RGB-D camera and a microphone array. Right: Mouth pose detection using the point cloud of the camera

from execution to execution. Based on our domain knowledge, we extracted the following features over time:

- **Pushing behavior**: *sound*(s), *force*(f), *approach distance*(k), and *approach angle*(k).[2] *Sound* is audio energy expressed in Eq. (16). *Force* is the magnitude of force $\mathbf{f}$ on the end-effector ($f = \|\mathbf{f}\|_2$). *Approach distance* is the Euclidean distance between the tool and an ARTag attached on a target object. *Approach angle* is the angular difference between the pushing tool of the robot and a line perpendicular to a door or lid plane.
- **Scooping behavior**: *sound*(s), *force*(f), *approach distance*(k), and *approach angle*(k). *Approach distance* is the distance between the tool and the bottom of a bowl location estimated by the kinematics of the robot. *Approach angle* is the angular difference between the tool and a line perpendicular to the opening plane of the bowl.
- **Feeding behavior**: *sound*(s), *force*(f), *joint torque*(f), and *approach distance*(k). *Force* is the directional component of the force vector along the spoon. *Joint torque* is the feedback torque of the first pan-joint, which corresponds to rotation of shoulder around the vertical axis, of the PR2.

The feature we used should be applicable to a wide variety of tasks. We clarified the correspondence of the features through modality analysis in Sect. 7.

To extract audio energy, we use the “Yaafe audio feature extraction toolbox” (Mathieu et al. 2010) to convert a frame s into a numeric value of energy $\mathcal{E}$ using the root mean square (RMS),

$$\mathcal{E} = \sqrt{\frac{\sum_{i=1}^{N_{\text{frame}}} (s(i)/I_{\max})^2}{N_{\text{frame}}}}, \tag{16}$$

where N_{frame} is the audio frame size, which varies from 1,024 to 4,096 and $I_{\max}$ is 32,768—the maximum value of a 16-bit signed integer format.

To extract the *approach distance* and *approach angle* in the feeding behavior, the robot continuously tracks a user’s mouth using the wrist-mounted RGB-D camera. After locating 2D facial landmarks using the dlib library (King 2009; Kazemi and Sullivan 2014), we converted landmarks into 3D based on depth information and estimated the position and orientation of the mouth, similar to the process in Schrer et al.’s 2015 work. We calculated *approach distance* by finding the Euclidean distance between the target and end-effector position. We calculated *approach angle* that is angular difference $2\arccos(q_1, q_2)$, where q_1 and q_2 are the orientations (i.e., quaternions) of the target and the end-effector, respectively.

Algorithm 1 shows how our system preprocesses the extracted feature vectors to be identical in scale, length, and offset. For each feature of data, let $\mathbf{V} \in \mathbb{R}^{N \times M_{\text{raw}}}$ be a set of feature vectors from both non-anomalous and anomalous executions, where N and M_{raw} are the number of samples and length, respectively. We first zero each feature vector $\mathbf{v} \in \mathbf{V}$ by subtracting the average of the first four elements, $\mathbf{v} = \mathbf{v} - \sum_{i=1}^{4} \mathbf{v}[i]/4$. This zeroing makes $\mathbf{v}$ starts from the same value as other vectors regardless calibration or white noise. Then, we scale each $\mathbf{v}$ between 0 and 1. To handle different sampling rates and match with actual anomaly check frequency in the PR2, we fit $\mathbf{v}$ with a B-Spline curve and resample it to a length M. In this paper, we used $M = 140$ for the robot-assisted feeding and 200 otherwise. This approximately matches the rate at which our execution monitor runs in real time. Finally, this preprocessing resulted in a sequence of tuples. For example, in the case of *sound* $\mathcal{E}$ and *force* f, the sequence of length t was $\{(f_1, \mathcal{E}_1), (f_2, \mathcal{E}_2), \ldots, (f_t, \mathcal{E}_t)\}$.

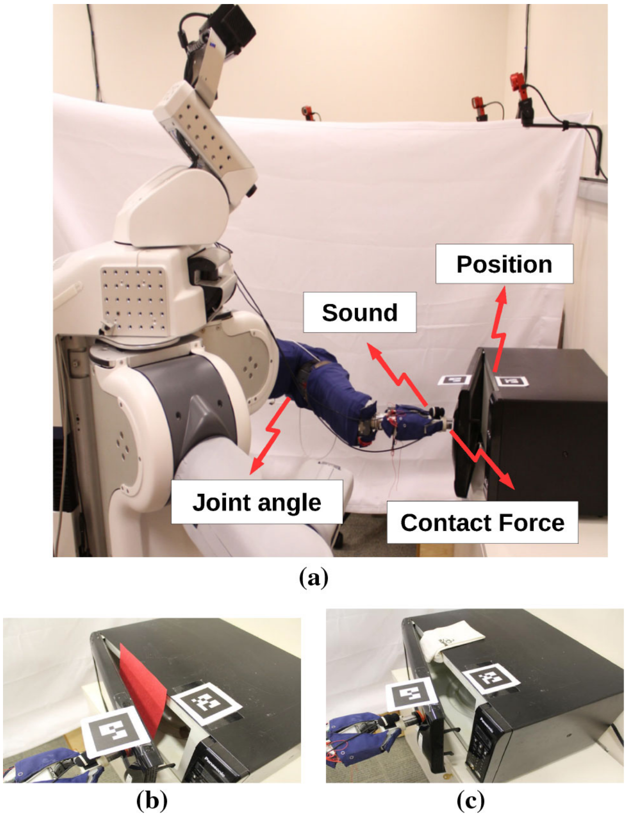
Algorithm 1: Preprocessing

```
input : $\mathbf{V} \in \mathbb{R}^{N \times M_{\text{raw}}}$, $M$
output: $\mathbf{V} \in \mathbb{R}^{N \times M}$
$v_{\max} \leftarrow \max(\mathbf{V})$, $v_{\min} \leftarrow \min(\mathbf{V})$;
for $i \leftarrow 1$ to $N$ do
    $\mathbf{v} \leftarrow \mathbf{V}[i] - \frac{\sum_{i=1}^{4} \mathbf{V}[i]}{4}$ ;            // Zeroing
    $\mathbf{v} \leftarrow \frac{\mathbf{v} - v_{\min}}{v_{\max} - v_{\min}}$ ;            // Scaling
    $\mathbf{v} \leftarrow \text{resample}(\text{curveFitting}(\mathbf{v}), M)$ ;   // Sampling
    $\mathbf{V}[i] \leftarrow \mathbf{v}$;
end
```

6.3 Experiment and evaluation procedures

For our experiments, we defined an anomalous execution as one that was inconsistent with typical executions. For exam-

[2] The symbols, f, s, and k, in the parentheses represent force, sound, and kinematic modalities, respectively.



(a)

(b)

(c)

Fig. 11 Sample scenes of pushing experiment with a microwave (B) door. **a** A scene of pushing experiment, **b**, **c** show two produced anomalies with paper and fabric obstacles, respectively

ple, even if the robot did scoop the yogurt in the scooping task, we classified the execution as anomalous if the user loudly commanded the robot to stop.

6.3.1 Pushing task

We collected pushing task data from two microwave ovens and a toolbox (see Fig. 11). The PR2 pushed each object with the instrumented tool and a predefined linear end-effector trajectory for an object-specific amount of time and then pulled its end effector back. To produce anomalous events, we placed the tool at an incorrect location from which it could not properly contact the target mechanism, fixed the mechanism to prevent movement, or blocked the mechanism using an obstacle such as a stack of paper, a fabric, a rubber pad, a ceramic bowl or a finger. During each execution, applied forces, sounds, end-effector poses, and object poses were measured via the sensing tools, the Kinect camera and the joint encoders. We recorded 54, 52, and 53 executions with non-anomalous data and 55, 56, and 51 executions with anomalous data in microwave white (Microwave (W)), microwave black (Microwave (B)), and toolbox closing tasks, respectively. The anomalous executions also included unintended failures while performing non-anomalous executions.

6.3.2 Robot-assisted feeding task

We collected data from a robot-assisted feeding task for which the PR2 executed a sequence of movements to scoop a spoon of yogurt from a bowl and then feed the yogurt to a human (see Fig. 12). For the scooping data collection, we recruited 5 able-bodied participants. We briefly trained them to use the system with one or two trials and then asked them to freely use the scooping and feeding behaviors in which they determined *non-anomalous* and *anomalous* labels for each behavior. We recorded 72 non-anomalous and 45 anomalous executions for the scooping behavior. To create anomalous executions, we asked each participant to intentionally fail when scooping yogurt in the following way:

– By having each participant push any part of the spoon, the bowl, or the arm of the PR2 during the scooping process
– By having the participant yell anything at any moment.

For the feeding data collection, we recruited 8 able-bodied participants [3] and collected data from 192 anomalous feeding execution (= 12 types × 2 executions per type × 8 participants) as well as 160 non-anomalous feeding execution. During non-anomalous feeding, we asked participants not to move their upper bodies except their necks to simulate the feeding task with motor impairments. To collect a broad range of anomalies, we selected 12 representative causes (see Fig. 13) from three groups of anomalies described in the literature (Ogorodnikova 2008; Vasic and Billard 2013),

– Human Error: a spoon miss, a spoon collision, a robot-body collision, aggressive eating, an anomalous sound, an unreachable mouth pose, and face occlusion.
– Engineering Error: a spoon miss, a spoon collision, and a controller failure.
– Environmental Error: an obstacle collision and a noise.

For each cause of anomalies listed, we briefly explained and showed a demonstration video, then we asked the participants to reproduce the anomaly twice at any time, magnitude, and direction during robot-assisted feeding.

6.4 Evaluation setup

6.4.1 Methods

In addition to our two proposed methods,

[3] Participants were 3 males and 5 females. Their age ranges from 19 to 35. They are either attending or have graduated college.

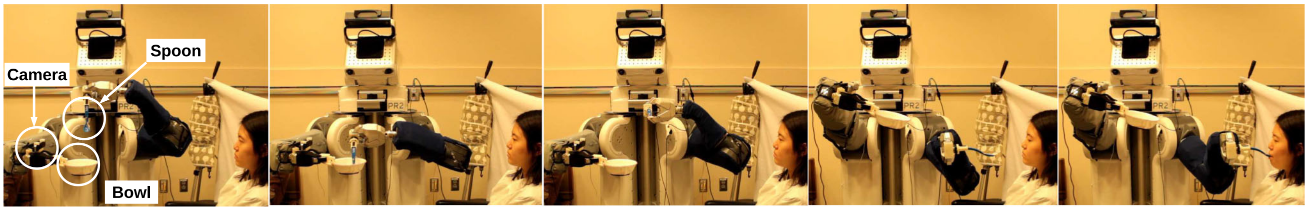


Fig. 12 The images show the entire scooping and feeding process

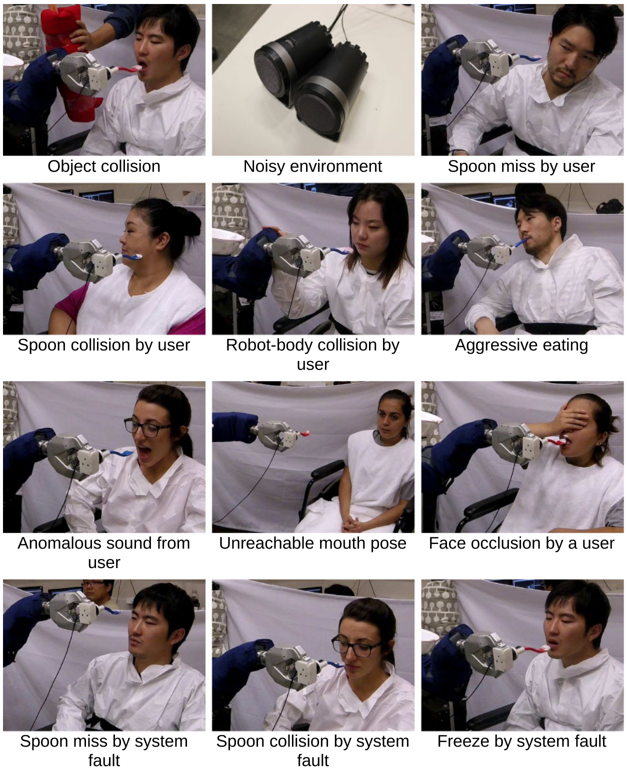

Object collision
Noisy environment
Spoon miss by user
Spoon collision by user
Robot-body collision by user
Aggressive eating
Anomalous sound from user
Unreachable mouth pose
Face occlusion by a user
Spoon miss by system fault
Spoon collision by system fault
Freeze by system fault

Fig. 13 Twelve representative cases for each anomalies in assistive feeding task

- HMM-D: A likelihood-based classifier with a clustering-based threshold.
- HMM-GP: A likelihood-based classifier with a regression-based threshold.

We implemented 6 base-line methods,

- RANDOM: This method randomly determines the existence of anomalies.
- OSVM: This method is an SVM-based detector trained with only negative training data (i.e., successful executions). Similar to Rodriguez et al.'s (2010) method, we move a sliding window (of size 10 in time) one step at a time over length of 140 or 200. After flattening and normalizing it with a zero mean and unit variance, we input the data to an OSVM with an RBF (Gaussian) kernel. We control its sensitivity by adjusting the number of support vectors.
- HMM-OSVM: We combined an HMM with an OSVM similar to Hung et al.'s (2010) method. Instead of the sliding window, we input an HMM-induced vector to the OSVM at each time step.
- HMM-C: A likelihood-based classifier that detects an anomaly when the change in log-likelihood exceeds a fixed threshold (Ocak and Loparo 2005).
- HMM-F: A likelihood-based classifier with a fixed threshold (Vaswani et al. 2005).
- HMM-KNN: A likelihood-based classifier with a clustering-based threshold. Unlike HMM-D, this method uses k-nearest neighbors to create clusters instead of RBFs.

6.4.2 Cross validation

For the pushing and scooping behaviors, we performed k-fold cross-validation, in which we randomly split non-anomalous and anomalous data into k folds, independently. To form the test data, we paired one fold from the non-anomalous data and one from the anomalous data, and used the remaining $k-1$ folds of the non-anomalous data for training. We repeated this process k^2 times so that each possible pair was used exactly once as test data. Note that we used $k = 3$ in this work. For the feeding behavior, we used leave-one-person-out cross validation for multiple subject data.

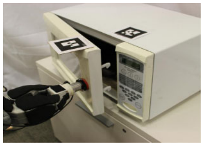
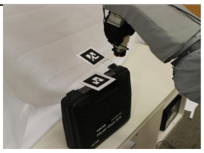
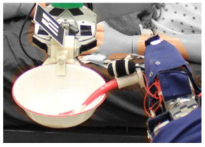
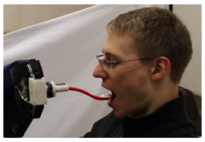
Training consisted of first fitting an HMM to the specific behavior and the particular object or human user. Using the output of the HMM, we then trained each classification method. In this study, we fixed the number of hidden states at 25 that is empirically determined and applied to all HMMs used in this paper. For the progress-based classifier, we used 25 RBFs, to match the number of hidden states. We added small random noise into the training data to provide variance.

7 Evaluation with two manipulation tasks

7.1 Examples of time-varying thresholds

The bottom graphs of Fig. 7 show our time-varying likelihood threshold from the clustering method. For the non-anomalous

Table 1 Detection performance (AUC) of our two methods and six baseline methods with 4 hand-engineered features from force, sound, and kinematic modalities

	IADL			ADL		
Method						Average
	Microwave (B)	Microwave (W)	Toolbox	Scooping	Feeding	
Random	0.4782	0.5001	0.4454	0.4750	0.5121	0.4822
OSVM	0.5023	0.6589	0.4277	0.6870	0.7238	0.5999
HMM-F	0.7764	0.6084	0.6735	0.6038	0.5055	0.6335
HMM-OSVM	0.7439	0.6436	0.6655	0.6478	0.7179	0.6837
HMM-C	0.8828	0.9405	0.4454	0.7183	0.7711	0.7516
HMM-KNN	0.9057	0.9853	0.7402	0.8249	0.8015	0.8515
HMM-D	0.9072	0.9935	**0.7796**	0.8293	0.8079	0.8627
HMM-GP	**0.9113**	**0.9938**	0.7654	**0.8608**	**0.8121**	**0.8687**

We show the highest AUC for each task in bold. Each task is categorized as either instrumental activities of daily life (IADL) or activities of daily life (ADL)

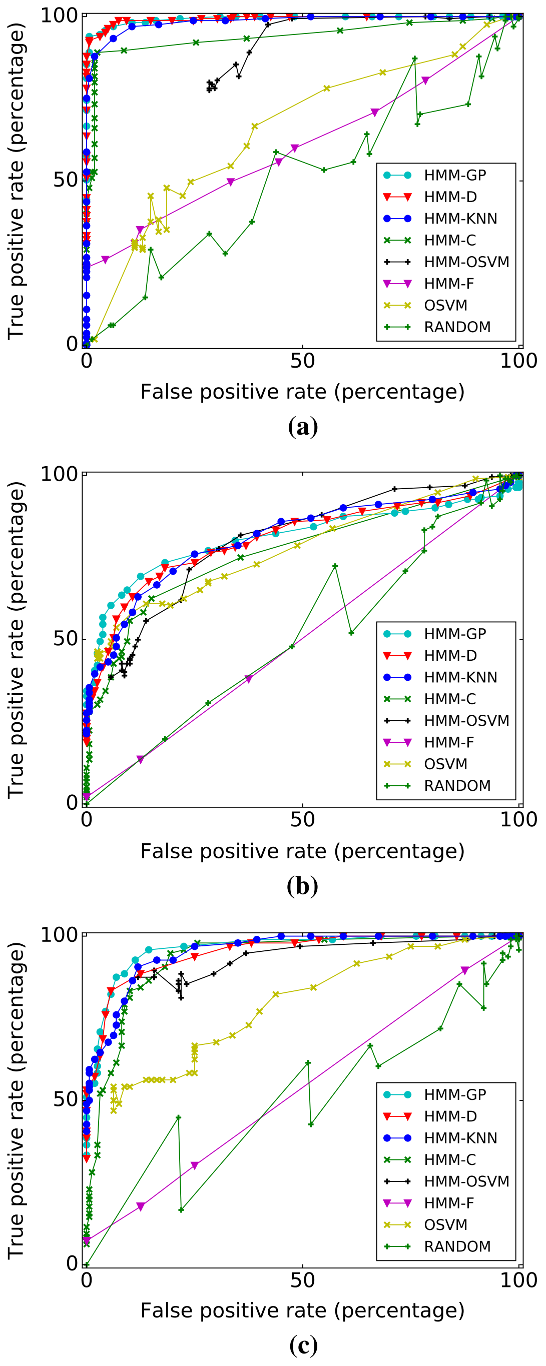
execution, the mean log-likelihood based on execution progress (solid red curve) moves in conjunction with the log-likelihood resulting from the ongoing trial (solid blue curve). The standard deviation based on execution progress (related to the dashed red curve) tends to increase over time. For the anomalous execution, the behavior failed to generate a sharp sound at the appropriate time and instead generated a lower magnitude sharp sound early in the behavior's execution in conjunction with lower forces than anticipated. As a result, the index 4 was not likely transitioned to a next index compared to the non-anomalous execution. The log-likelihood went below the threshold early on in the execution, triggering the detection of an anomaly. This illustrates that execution progress is helpful to determine a tighter decision boundary than a fixed threshold.

7.2 Performance comparison with baseline methods

AUC We compared our HMM-D and HMM-GP to baseline methods. Table 1 shows that HMM-GP resulted in higher AUC than others over the five tasks where we show the highest AUC for each task in bold. HMM-GP showed the highest AUC in four tasks with a maximum of 0.9938. HMM-D shows similar but slightly lower performance, but it outperformed HMM-KNN. This result shows the training with the time-based RBFs was beneficial. In addition, the feeding task's AUCs, computed by the leave-one-person-out cross validation, demonstrates our that an execution monitor can generalize to new people. In Fig. 14, the receiver operating characteristic (ROC) curves also show HMM-GP and HMM-D result in higher true positive rates (TPR) than the other baseline methods for comparable false positive rates (FPR). The performance was high with the microwave task. However, the feeding task performance was highly dependent on the types of anomalies. To illustrate this, Fig. 14 shows an ROC curve with all 12 anomalies and an ROC curve that only considers the 6 best detected anomalies. In general, this highlights the importance of using modalities and features suitable for detecting important types of anomalies. In our evaluation, we attempted to use a wide variety of realistic anomalies, some of which were challenging for the system.

Another interesting result is that OSVM showed lower performance than others except RANDOM. The reason could be that OSVM cannot sufficiently convey contextual information from the past since it uses a sliding window. HMM-OSVM's performance was substantially better than OSVM, but its performance was still lower than our proposed methods. The reason could be that the highly nonlinear RBF kernel in HMM-OSVM led to overfitting and generated false alarms. In addition, 6 out of 8 methods we evaluated had their worst performance (AUC) on the toolbox pushing task, and no method performed particularly well. This is most likely due to the high variability in the signals (see Fig. 4b). The especially poor performance of HMM-C suggests that the likelihood values produced by the HMM could exhibit large changes from time step to time step in the toolbox pushing task. Note that rounding made the AUC rates of RND and HMM-C appear the same. The raw AUC rates of RND and HMM-C were 0.445410449295 and 0.445389896, respectively.

Through evaluation, in terms of the amount of data we had, HMM-GP was well matched to the pushing tasks, which had 50 or fewer time series when considering cross validation. For the feeding task, HMM-GP was not as well matched due



(a)

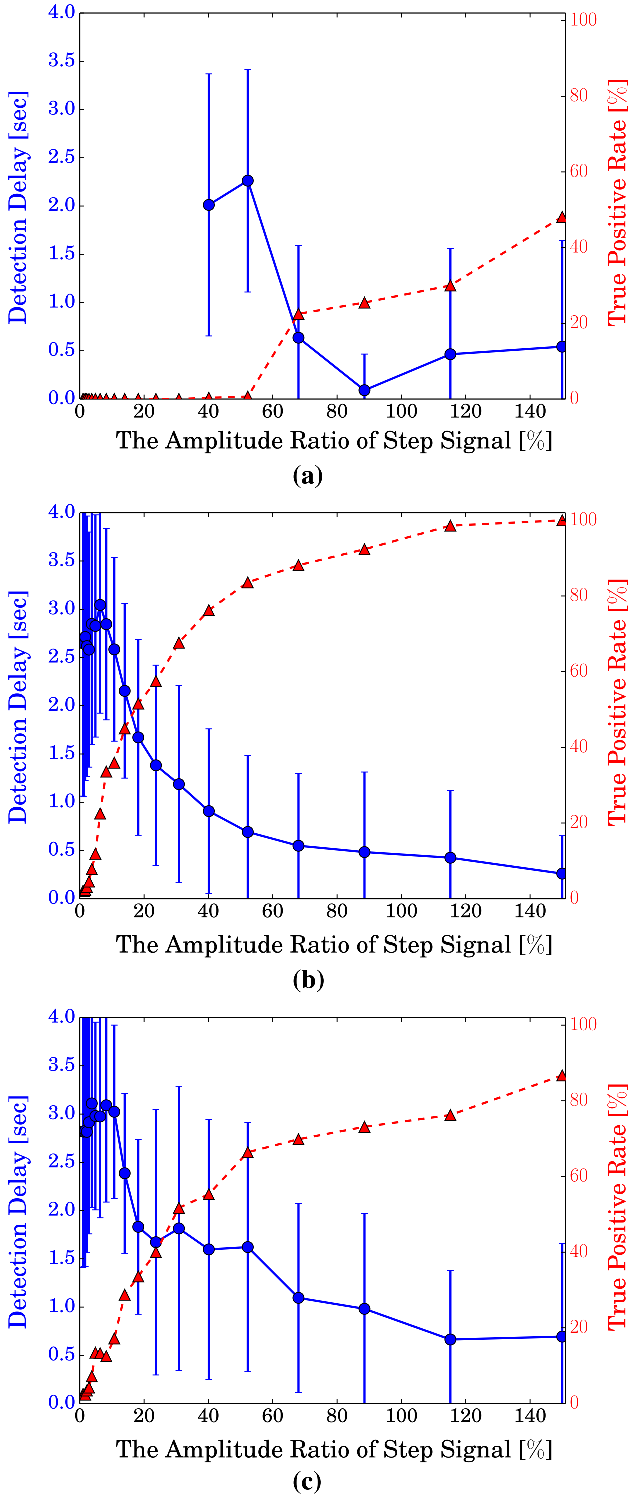
(b)

(c)

Fig. 14 Receiver operating characteristic (ROC) curves to evaluate the performance of our anomaly detectors versus six baseline detectors in the pushing task with the microwave (W) and the feeding task. **a** Pushing task with a microwave (W) and 55 anomalies, **b** feeding task with 12 causes of anomalies, **c** feeding task with top 6 causes of anomalies presented in Fig. 16

to the larger amount of data. Consequently, we performed random sub-sampling of the data, which performed well in practice.

Detection Delay Another performance measure for an execution monitor is the detection delay. Figure 15 presents a



(a)

(b)

(c)

Fig. 15 Comparison between the detection delays and the true positive rates on simulated anomaly signals introduced in the assistive feeding task. We added a step signal to a feature of non-anomalous data in which the amplitude of a signal ranges from 1 to 150% of the maximum amplitude of observed anomaly signals in real experiments. We take only positive delays into account. **a** HMM-F, **b** HMM-D, **c** HMM-GP

Table 2 Detection performance over modalities with HMM-GP

	Microwave (B)	Microwave (W)	Toolbox	Scooping	Feeding	Average
s	0.6839	0.7353	0.5496	0.5168	0.6225	0.6216
f	0.5058	0.7024	0.4686	0.8123	0.6441	0.6266
f,s	0.6672	0.8288	0.5537	0.7178	0.7632	0.7061
k	0.7703	0.8199	0.6193	0.8212	0.6355	0.7332
s,k	0.8922	0.9759	0.6524	0.7999	0.6173	0.7875
f,k	0.8665	0.9218	0.6743	**0.9568**	0.7458	0.8330
f,s,k	**0.9113**	**0.9938**	**0.7654**	0.8608	**0.8121**	**0.8687**

Modalities *f*, *s*, and *k* represent force, sound, and kinematic modalities, respectively. (e.g., (f, s, k) indicates the use of all modalities used in Table 1)
The highest values of AUC are highlighted in bold

comparison between the detection delay and the true positive rate changes of our methods and the HMM-F method in the feeding task. We generated various simulated anomaly data by adding a step signal from a magnitude of 1–150% of the highest real anomaly signals to a randomly selected feature of the non-anomalous data at random times. The simulated anomaly data corresponds to "spoon collision," "arm collision," "spoon miss by a user," and "anomalous sound." The HMM-D and -GP methods resulted in shorter detection delays with higher true positive rates. In particular, our methods detected small anomalies $(\leq 10\%)$ within 3.0 second. On the other hand, HMM-F was able to detect the large anomalies $(> 10\%)$ with longer detection delays and lower true positive rates (TPR). In this process, the higher sensitivity (i.e., TPR) we used to select the parameter of classifiers, the shorter the detection delay we observed.

In this evaluation, we tuned the three detectors to have the same, small false positive rate (FPR) given a test set. Similar to Neyman-Pearson criteria (Fukunaga 2013), we bound the range of false positive rates and maximized the true positive rates. However, due to limited data, the detectors could not achieve arbitrary false positive rates. Consequently, we exhaustively searched the false positive rates achievable by all three detectors. We found that $1.0 \pm 1.0\%$ is the range of the commonly achievable smallest false positive rate.

The detection delays tend to become longer and the true positive rates tend to become lower as the anomalies decrease in magnitude. This is an important consideration for assistive robots, since both the detection delay and the true positive rate are important for anomaly detection. For some types of errors, detection after a 3 second delay might be too late for the robot to make a meaningful corrective action. True positive rates that are too high could lead to people not using a system and finding it annoying. Note that we did not evaluate other detector tunings, so we can not be certain that these results would hold for other tunings. However, we would expect the general trends for subtle anomalies to hold.

7.3 Multimodality

Multimodality versus Unimodality We investigated if multiple modalities improve the detection performance of our monitoring system. Table 2 shows AUC over combinations of modalities (i.e., force, sound, kinematics[4]) with HMM-GP. The use of all three modalities achieved the best performance in four of the five tasks. The use of one, two, or three modalities showed maximum 0.7332, 0.8830, and 0.8687 of the averaged AUC, respectively. The AUC indicated that multi-modalities were substantially more effective than unimodalities in this detection. However, it does not indicate that more modalities always enhance detection performance. In the scooping task, including sound as a modality reduced the detector's performance. It seems likely that the sound recordings for this task varied in a manner that was unrelated to the anomalies we tested, and thus did not improve anomaly detection and instead resulted in false positives.

Effective Modalities Effectiveness of modalities varies according to the type of anomaly. Figure 16 shows the TPR matrix in which the x- and y-axes represent input modalities and tested anomaly classes, respectively. We ordered anomaly classes with respect to a similar TPR distribution over the three unimodalities using density-based spatial clustering of applications with noise (DBSCAN). The clustering resulted in four groups of anomalies (red brackets and numbers). The first three groups show that the detection of anomalies primarily depends on a single modality. We also observed that multimodality improved detection performance. For example, the force-kinematics modality combination yielded higher TPR in three of four anomalies in the first group. All three combinations of modalities also enabled our system to detect the broadest range of anomalies. However, the anomalies in the last group were not easily detectable. In particular, "face occlusion" and "freeze" were

[4] Kinematic modality refers to the task-kinematic input measured by the encoder and vision sensors (i.e, relative distance or orientation between the PR2 and a target object or human).

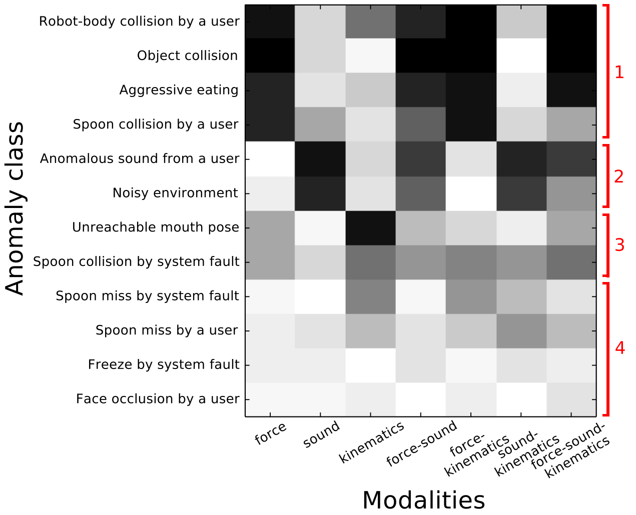


Fig. 16 True positive rate per anomaly class and modalities in the assistive feeding task. Each column shows the true positive rate distribution of an HMM-GP from a combination of modalities. We tuned the threshold of the detector to produce an 8 ± 0.7% false positive rate (FPR) that is the commonly achievable smallest value by the 7 set of modalities. Black and white regions represent 100 and 0% true positive rates, respectively. Red brackets and numbers show four clusters found via DBSCAN (Color figure online)

hardly detectable from any combination of modalities that we used. We expect that additional features or modalities might help to detect these anomalies. For example, modalities that monitor the internal state of the robot, including the activity of computational processes, might better detect some types of system freezes and faults. Note that we only produced these results for specific detector tunings, so the details could change with other tunings. However, we would expect the associations between particular modalities and particular types of anomalies to hold across tunings, as well as the existence of anomalies that are difficult to detect even with all of the modalities available.

8 Conclusion

We presented an execution monitoring system for multimodal anomaly detection during assistive robot manipulation. Our system modeled multimodal sensory readings associated with non-anomalous executions with a multivariate Gaussian HMM. Our approach takes advantage of similar patterns of change that tend to occur in hidden-state distributions and log-likelihoods during non-anomalous executions. Our system detects an anomaly when a log-likelihood is lower than a varying rejection threshold. Our methods varied the threshold with respect to an estimated execution progress introduced in our previous work (Park et al. 2016a). To estimate the threshold parameters, we introduced two methods, a clustering-based classifier (HMM-D) and a regression-based classifier (HMM-GP). We evaluated our methods with a PR2 robot performing object pushing and robot-assisted feeding tasks. Our methods outperformed 6 baseline methods from the literature by providing higher AUCs and shorter detection delays. We also showed that the detection of each anomaly tends to depend on a distinct set of modalities and that the use of multiple modalities was beneficial for anomaly detection.

Acknowledgements We thank Youkeun Kim, Zackory Erickson, Ariel Kapusta, Chansu Kim, and Jane Chisholm for their assistance throughout this project. This work was supported in part by NSF Awards IIS-1150157, EFRI-1137229, and NIDILRR Grant 90RE5016-01-00 via RERC TechSAge. Dr. Kemp is a cofounder, a board member, an equity holder, and the CTO of Hello Robot, Inc., which is developing products related to this research. This research could affect his personal financial status. The terms of this arrangement have been reviewed and approved by Georgia Tech in accordance with its conflict of interest policies.

References

Ando, S., Thanomphongphan, T., Hoshino, D., Seki, Y., & Suzuki, E. (2011). ACE: Anomaly clustering ensemble for multi-perspective anomaly detection in robot behaviors. In *Proceedings of the international conference on data mining* (pp. 1–12). SIAM.

Angelov, V. P., Giglio, C., Guardiola, C., Lughofer, E., & Lujan, J. M. (2006). An approach to model-based fault detection in industrial measurement systems with application to engine test benches. *Measurement Science and Technology*, *17*(7), 1809.

Argall, B. D. (2016). Modular and adaptive wheelchair automation. In M. A. Hsieh, O. Khatib, & V. Kumar (Eds.), *Experimental robotics* (pp. 835–848). Springer.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. New York, NJ: Springer.

Bittencourt, A. C., Saarinen, K., Sander-Tavallaey, S., Gunnarsson, S., & Norrlöf, M. (2014). A data-driven approach to diagnostics of repetitive processes in the distribution domain-applications to gearbox diagnostics in industrial robots and rotating machines. *Mechatronics*, *24*(8), 1032–1041.

Blank, S., Pfister, T., & Berns, K. (2011). Sensor failure detection capabilities in low-level fusion: a comparison between fuzzy voting and kalman filtering. In *IEEE international conference on robotics and automation (ICRA)* (pp 4974–4979). IEEE.

Bouguerra, A., Karlsson, L., & Saffiotti, A. (2008). Monitoring the execution of robot plans using semantic knowledge. *Robotics and Autonomous Systems*, *56*(11), 942–954.

Brambilla, D., Capisani, L. M., Ferrara, A., & Pisu, P. (2008). Fault detection for robot manipulators via second-order sliding modes. *IEEE Transactions on Industrial Electronics*, *55*(11), 3954–3963. https://doi.org/10.1109/TIE.2008.2005932. ISSN 0278-0046.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882, http://dl.acm.org/citation.cfm?id=1541880.1541882, ISSN 03600300.

Chen, T. L., Ciocarlie, M., Cousins, S., Grice, P., Hawkins, K., Hsiao, et al. (2013). Robots for humanity: Using assistive robots to empower people with disabilities. *IEEE Robotics & Automation Magazine*, *20*(1), 30–39.

Chu, V., McMahon, I., Riano, L., McDonald, C. G., He, Q., Perez-Tejada, J. M., et al. (2013). Using robotic exploratory procedures to learn the meaning of haptic adjectives. In *IEEE international* *conference on robotics and automation (ICRA)* (pp. 3048–3055). IEEE.

Ciocarlie, M., Hsiao, K., Leeper, A., & Gossow, D. (2012). Mobile manipulation through an assistive home robot. In *IEEE/RSJ international conference on intelligent robots and systems*. IEEE.

Clifton, D. A., Hugueny, S., & Tarassenko, L. (2011). Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems*, *65*(3), 371–389.

Copilusi, C., Kaur, M., & Ceccarelli, M. (2015a). *Lab experiences with LARM clutched arm for assisting disabled people* (pp. 603–611). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-09411-3_64. ISBN: 978-3-319-09411-3.

Copilusi, C., Kaur, M., & Ceccarelli, M., (2015b). Lab experiences with larm clutched arm for assisting disabled people. In *New trends in mechanism and machine science* (pp. 603–611). Springer.

Cressie, N. A. C. (1993). *Statistics for spatial data*. New York, CH: Wiley series in probability and mathematical statistics, Wiley. ISBN: 978-0-471-00255-0. http://opac.inria.fr/record=b1085822.

Dames, P. M., Schwager, M., & Rus, D. (2016). Active magnetic anomaly detection using multiple micro aerial vehicles. *IEEE Robotics and Automation Letters*, *1*(1), 153–160. https://doi.org/10.1109/LRA.2015.2511444. ISSN: 2377-3766.

Di Lello, E., Klotzbucher, M., De Laet, T., & Bruyninckx, H. (2013). Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5827–5833). IEEE.

Dua, S., & Du, X. (2011). *Data mining and machine learning in cybersecurity*. London: CRC Press.

Eclipse Automation. (2016). Meet obi, a robot that helps disabled individuals eat unassisted. https://meetobi.com/. Accessed 15 July 2017.

Fagogenis, G., Carolis, V. D., & Lane, D. M. (2016). Online fault detection and model adaptation for underwater vehicles in the case of thruster failures. In *IEEE international conference on robotics and automation*. IEEE.

Fiala, M. (2005). Artag, a fiducial marker system using digital techniques. In *IEEE computer society conference on computer vision and pattern recognition* (Vol. 2, pp. 590–596). IEEE.

Fujii, H., Yamashita, A., & Asama, H., (2016). Defect detection with estimation of material condition using ensemble learning for hammering test. In *IEEE international conference on robotics and automation*. IEEE.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. London: Academic press.

Gehring, J., Miao, Y., Metze, F., & Waibel, A. (2013). Extracting deep bottleneck features using stacked auto-encoders. In *Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3377–3381). IEEE.

Graf, B., Reiser, U., Hägele, M., Mauz, K., & Klein, P. (2009). Robotic home assistant care-o-bot® 3-product vision and innovation platform. In *IEEE workshop on advanced robotics and its social impacts (ARSO)* (pp. 139–144). IEEE.

Haidu, A., Kohlsdorf, D., & Beetz, M., (2015) Learning action failure models from interactive physics-based simulations. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5370–5375). IEEE.

Hawkins, K. P., Grice, P. M., Chen, T. L., King, C.-H., & Kemp, C. C. (2014) Assistive mobile manipulation for self-care tasks around the head. In *IEEE symposium on computational intelligence in robotic rehabilitation and assistive technologies* (pp. 16–25). IEEE.

Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern recognition*, *40*(3), 863–874.

Hung, Y.-X., Chiang, C.-Y., Hsu, S.J., & Chan, C.-T., (2010). Abnormality detection for improving elders daily life independent. In *International conference on smart homes and health telematics* (pp. 186–194). Springer.

Jain, A., & Kemp, C. C. (2013). Improving robot manipulation with data-driven object-centric models of everyday forces. *Autonomous Robots*, *35*(2–3), 143–159.

Jiménez Villarreal, J., & Ljungblad, S. (2011) Experience centred design for a robotic eating aid. In *Proceedings of the 6th international conference on human–robot interaction* (pp. 155–156). ACM.

Kappler, D., Pastor, P., Kalakrishnan, M., Wuthrich, M., & Schaal, S. (2015). Data-driven online decision making for autonomous manipulation. In *Proceedings of robotics: Science and systems*.

Kazemi, V., & Sullivan, J. (2014) One millisecond face alignment with an ensemble of regression trees. In *IEEE conference on computer vision and pattern recognition* (pp. 1867–1874). https://doi.org/10.1109/CVPR.2014.241.

Kemp, C. C., Edsinger, A., & Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, *14*(1), 20–29.

Ketabdar, H., Vepa, J, Bengio, S., & Bourlard, H. (2006) Using more informative posterior probabilities for speech recognition. In *IEEE international conference on acoustics speech and signal processing proceedings* (Vol. 1, pp. 1). IEEE.

Khan, S. S., Karg, M. E., Hoey, J., & Kulic, D. (2012). Towards the detection of unusual temporal events during activities using HMMs. In *Proceedings of the ACM conference on ubiquitous computing* (pp. 1075–1084). ACM.

Kim, D. J., Wang, Z., Paperno, N., & Behal, A. (2014). System design and implementation of UCF-MANUS—an intelligent assistive robotic manipulator. *IEEE/ASME transactions on mechatronics*, *19*(1), 225–237. https://doi.org/10.1109/TMECH.2012.2226597. ISSN: 1083-4435.

King, C. H., Chen, T. L., Jain, A., & Kemp, C. C. (2010). Towards an assistive robot that autonomously performs bed baths for patient hygiene. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 319–324). https://doi.org/10.1109/IROS.2010.5649101.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.

Lee, H., Hwang, B., & Cho, S. (2002). Analysis of novelty detection properties of autoassociative mlp. *Journal of Korean Institute of Industrial Engineers*, *28*(2), 147–161.

Leidner, D., Dietrich, A., Beetz, M., & Albu-Schäffer, A. (2016). Knowledge-enabled parameterization of whole-body control strategies for compliant service robots. *Autonomous Robots*, *40*(3), 519–536. https://doi.org/10.1007/s10514-015-9523-3. ISSN: 1573-7527.

Lepora, N. F., Pearson, M. J., Mitchinson, B., Evans, M., Fox, C., Pipe, A., Gurney, K., & Prescott, T. J. (2010) Naive bayes novelty detection for a moving robot with whiskers. In *IEEE international conference on robotics and biomimetics (ROBIO)* (pp. 131–136). IEEE.

Lühr, S., Venkatesh, S., West, G., & Bui, H. H. (2004). Explicit state duration HMM for abnormality detection in sequences of human activity. In *Pacific rim international conference on artificial intelligence* (pp. 983–984). Springer.

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P., (2015). Long short term memory networks for anomaly detection in time series. In *23rd European symposium on artificial neural networks, computational intelligence and machine learning* (p. 89).

Marcolino, F., & Wang, J. (2013). Detecting anomalies in humanoid joint trajectories. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2594–2599). IEEE.

Markou, M., & Singh, S. (2003). Novelty detection: Areviewpart 1: statistical approaches. *Signal Processing*, *83*(12), 2481–2497.

Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. (2010). Yaafe, an easy to use and efficient audio feature extraction software. In

Proceedings of the 11th international conference on music information retrieval (ISMIR (pp. 441–446).

Mealtime Partners. (2017). *Specializing in assistive dining and drinking equipment*. http://www.mealtimepartners.com/. Accessed July 15, 2017.

Mendoza, J. P., Veloso, M., & Simmons, R. (2014). Focused optimization for online detection of anomalous regions. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3358–3363). IEEE.

Morris, B. T., & Trivedi, M. M. (2008). Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. In *IEEE fifth international conference on advanced video and signal based surveillance (AVSS)* (pp. 154–161). IEEE.

Nguyen, H., Anderson, C., Trevor, A., Jain, A., Xu, Z., & Kemp, C. C. (2008). El-e: An assistive robot that fetches objects from flat surfaces. In *Robotic helpers, international conference on human–robot interaction*.

Niekum, S., Osentoski, S., Atkeson, C. G., & Barto, A. G., (2014). Learning articulation changepoint models from demonstration. In *Robotics science and systems (RSS) workshop on learning plans with context from human signals*.

Ocak, H., & Loparo, K. A. (2005). HMM-based fault detection and diagnosis scheme for rolling element bearings. *Journal of Vibration and Acoustics*, *127*(4), 299–306.

Ogorodnikova, O. (2008). Methodology of safety for a human robot interaction designing stage. In *Conference on human system interactions* (pp. 452–457). IEEE.

Papageorgiou, X. S., Tzafestas, C. S., Maragos, P., Pavlakos, G., Chalvatzaki, G., Moustris, G., et al. (2014). Advances in intelligent mobility assistance robot integrating multimodal sensory processing. In *International conference on universal access in human–computer interaction* (pp. 692–703). Springer.

Park, D., Erickson, Z., Bhattacharjee, T., & Kemp, C. C. (2016a). Multimodal execution monitoring for anomaly detection during robot manipulation. In *IEEE international conference on robotics and automation*. IEEE.

Park, D., Kim, Y. K., Erickson, Z., & Kemp, C. C. (2016b). 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*.

Pastor, P., Kalakrishnan, M., Chitta, E., Theodorou, S., & Schaal, S. (2011). Skill learning and task outcome prediction for manipulation. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3828–3834). IEEE.

Patterson Medical. Meal buddy, (2017). http://pattersonmedical.com/. Accessed: July 15, 2017.

Perduca, V., & Nuel, G. (2013). Measuring the influence of observations in HMMs through the kullback-leibler distance. *IEEE Signal Processing Letters*, *20*(2), 145–148.

Pettersson, O. (2005). Execution monitoring in robotics: A survey. *Robotics and Autonomous Systems*, *53*(2), 73–88.

Pimentel, M. A. F., Clifton, L., David, A., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE* (pp. 257–286).

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning* (pp. 63–71). Springer.

Rasmussen, C. E., Kuss, M., et al. (2003). Gaussian processes in reinforcement learning. In *Advances in neural information processing systems* (Vol. 4, p. 1).

Rodriguez, A., Bourne, D., Mason, M., Rossano, G. F., & Wang, J., (2010). Failure detection in assembly: Force signature analysis. In *IEEE conference on automation science and engineering (CASE)* (pp. 210–215). IEEE.

Rodriguez, A., Mason, M. T., Srinivasa, S., Bernstein, M., & Zirbel, A. (2011). Abort and retry in grasping. In *IEEE international conference on intelligent robots and systems (IROS)*.

Sakaguchi, T., Yokoi, K., Ujiie, T., Tsunoo, S., & Wada, K. (2009). Design of common environmental information for door-closing tasks with various robots. *International Journal of Robotics and Automation*, *24*(3), 203.

Schliep, A., Rungsarityotin, W., & Georgi, B. (2004). General hidden markov model library. http://www.ghmm.org/.

Schrer, S., Killmann, I., Frank, B., Vlker, M., Fiederer, L., Ball, T., & Burgard, W. (2015). An autonomous robotic assistant for drinking. In *IEEE international conference on robotics and automation (ICRA)* (pp. 6482–6487). https://doi.org/10.1109/ICRA.2015.7140110.

Serdio, F., Lughofer, E., Pichler, K., Buchegger, T., & Efendic, H. (2014). Residual-based fault detection using soft computing techniques for condition monitoring at rolling mills. *Information Sciences*, *259*, 304–320.

Silvrio, J., Rozo, L., Calinon, S.,& Caldwell, D. G. (2015). Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 464–470). https://doi.org/10.1109/IROS.2015.7353413.

Simanek, J., Kubelka, V., & Reinstein, M. (2015). Improving multimodal data fusion by anomaly detection. *Autonomous Robots*, *39*(2), 139–154.

Snelson, E., & Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, *18*, 1257.

Sölch, M., Bayer, J., Ludersdorfer, M., & van der Smagt, P. (2016). Variational inference for on-line anomaly detection in high-dimensional time series. arXiv preprint arXiv:1602.07109.

Song, W.-K., & Kim, J. (2012). *Novel assistive robot for self-feeding*. Rijeka: INTECH Open Access Publisher.

Suarez, A., Heredia, G., & Ollero, A., (2016). Cooperative sensor fault recovery in multi-UAV systems. In *IEEE international conference on robotics and automation*. IEEE.

Suetani, H., Ideta, A. M., & Morimoto, J. (2011). Nonlinear structure of escape-times to falls for a passive dynamic walker on an irregular slope: Anomaly detection using multi-class support vector machine and latent state extraction by canonical correlation analysis. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 2715–2722). IEEE.

Sukhoy, V., Georgiev, V., Wegter, T., Sweidan, R., & Stoytchev, A., (2012). Learning to slide a magnetic card through a card reader. In *IEEE international conference on robotics and automation (ICRA)* (pp. 2398–2404). IEEE.

Takahashi, Y., & Suzukawa, S. (2006). Easy human interface for severely handicapped persons and application for eating assist robot. In *IEEE international conference on mechatronics* (pp. 225–229). https://doi.org/10.1109/ICMECH.2006.252529.

Topping, M. (2002). An overview of the development of handy 1, a rehabilitation robot to assist the severely disabled. *Journal of Intelligent and Robotic Systems*, *34*(3), 253–263.

Vasic, M., & Billard, A. (2013). Safety issues in human–robot interactions. In *IEEE international conference on robotics and automation (ICRA)* (pp. 197–204). IEEE.

Vaswani, N., Roy-Chowdhury, A. K., & Chellappa, R. (2005). Shape activity: A continuous-state HMM for moving/deforming shapes with application to abnormal activity detection. *IEEE Transactions on Image Processing*, *14*(10), 1603–1616.

Wakita, Y., Yoon, W. K., & Yamanobe, W. K., (2012). User evaluation to apply the robotic arm rapuda for an upper-limb disabilities patient's daily life. In *IEEE international conference on robotics and biomimetics (ROBIO)* (pp. 1482–1487). https://doi.org/10.1109/ROBIO.2012.6491178.

Warrender, C., Forrest, S., & Pearlmutter, B. (1999). Detecting intrusions using system calls: Alternative data models. In *Proceedings of the IEEE symposium on security and privacy* (pp. 133–145). IEEE.

Wiener, J. M., Raymond, W., Hanley, J., Clark, R., Nostrand, V., & Joan, F. (1990). Measuring the activities of daily living: Comparisons across national surveys. *Journal of Gerontology*, *45*(6), S229–S237.

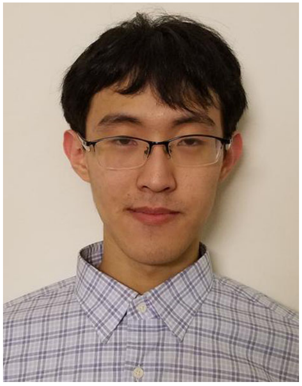
Williams, G., Baxter, R., He, H., Hawkins, S., & Gu, L. (2002). A comparative study of RNN for outlier detection in data mining. In *Proceedings of IEEE international conference on data mining, ICDM* (pp. 709–712). IEEE.

Yamazaki, K., Ueda, R., Nozawa, S., Mori, Y., Maki, T., Hatao, N., et al. (2009). A demonstrative research for daily assistive robots on tasks of cleaning and tidying up rooms. In *First international symposium on quality of life technology (June 2009)*.

Yeung, D.-Y., & Ding, Y. (2003). Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition*, *36*(1), 229–243.

Yu, D., & Seltzer, M L., (2011). Improved bottleneck features using pretrained deep neural networks. In *INTERSPEECH* (Vol. 237, p. 240).

Daehyung Park received the B.E. degree from the Department of Systems Science, Osaka University, Osaka, Japan, and the M.S. degree from the Department of Computer Science, University of Southern California, Los Angeles, USA. He is currently a Robotics Ph.D. Student at the Georgia Institute of Technology in Atlanta, USA. He also worked as a Robot Engineer with the Mechatronics Center, Samsung Electronics Inc., Suwon, Korea. His research interests include assistive manipulation, haptic perception, and machine learning.



Hokeun Kim is a Computer Science undergraduate student at the Georgia Institute of Technology. His interests include artificial intelligence, computer vision, and machine learning.

Charles C. Kemp is an Associate Professor at the Georgia Institute of Technology in the Department of Biomedical Engineering. He also has adjunct appointments in the School of Interactive Computing and the School of Electrical and Computer Engineering. He earned a doctorate in Electrical Engineering and Computer Science (2005), an MEng, and BS from MIT. In 2007, he founded the Healthcare Robotics Lab at Georgia Tech (http://healthcare-robotics.com), which focuses on mobile manipulation and human-robot interaction. He is an active member of the Institute for Robotics and Intelligent Machines (IRIM@GT) and Georgia Tech's multi-disciplinary Robotics Ph.D. program. He has received the 3M Non-tenured Faculty Award, the Georgia Tech Research Corporation Robotics Award, and the NSF CAREER award.