A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder

Daehyung Park, Yuuna Hoshi, and Charles C. Kemp

Abstract—**The detection of anomalous executions is valuable for reducing potential hazards in assistive manipulation. Multimodal sensory signals can be helpful for detecting a wide range of anomalies. However, the fusion of high-dimensional and heterogeneous modalities is a challenging problem for model-based anomaly detection. We introduce a long short-term memory-based variational autoencoder (LSTM-VAE) that fuses signals and reconstructs their expected distribution by introducing a progress-based varying prior. Our LSTM-VAE-based detector reports an anomaly when a reconstruction-based anomaly score is higher than a state-based threshold. For evaluations with 1555 robot-assisted feeding executions, including 12 representative types of anomalies, our detector had a higher area under the receiver operating characteristic curve of 0.8710 than 5 other baseline detectors from the literature. We also show the variational autoencoding and state-based thresholding are effective in detecting anomalies from 17 raw sensory signals without significant feature engineering effort.**

Index Terms—**Failure detection and recovery, deep learning in robotics and automation, assistive robots.**

I. Introduction

PEOPLE with disabilities often need physical assistance from caregivers. Robots can provide assistance for activities of daily living such as robot-assisted feeding [1] and shaving [2]. However, its structural complexity, task variability, and sensor uncertainty may result in failures. A lack of detection systems for the failures may also lower the usage of robots due to potential failure cost. The detection of an anomalous task execution (i.e., anomaly) can help to prevent or reduce potential hazards in the assistance by recognizing and stopping in highly unusual situations.

Anomaly detection is a method to identify when the current execution differs from typical successful experiences (i.e., non-anomalous executions). Researchers often use a one-class classifier trained with non-anomalous executions. Some classifiers have difficulty using the high-dimensional multimodal sensory data that can be readily acquired with modern robots. Our previous work used 4 hand-engineered, manually selected features from 3 modalities for a likelihood-based classifier using hidden Markov models (HMM) [3], [4]. Lower-dimensional representations such as these can lose information relevant to anomaly detection. Creating useful hand-crafted features can also involve significant engineering effort or domain expertise.

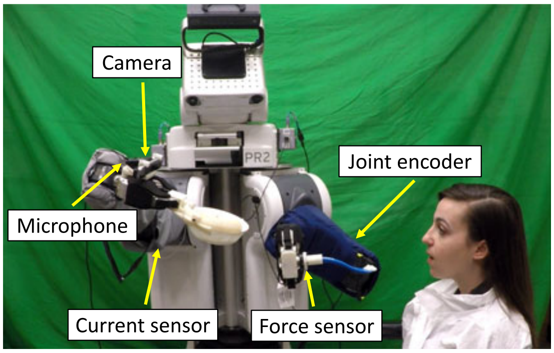

Fig. 1. Robot-assisted feeding system. A PR2 robot detects anomalous feeding executions collecting 17 sensory signals from 5 types of sensors.

An alternative solution is reconstruction-based detection by applying a reconstruction error as an anomaly score. Researchers often use an autoencoder (AE) to compress and reconstruct high dimensional input given non-anomalous training data. The idea behind this detection is that an AE cannot reconstruct unforeseen patterns of anomalous data well compared to foreseen non-anomalous data. In addition to the reconstruction error, a variational autoencoder (VAE) can compute the reconstruction log-likelihood of the input modeling the underlying probability distribution of data. Both AE and VAE based networks can be combined with time-series modeling approaches such as recurrent neural networks (RNN) including long short-term memory (LSTM) networks.

In this letter, we introduce an LSTM-based VAE (LSTM-VAE) for multimodal anomaly detection. For encoding, an LSTM-VAE projects multimodal observations and their temporal dependencies at each time step into a latent space using serially connected LSTM and VAE layers. For decoding, it estimates the expected distribution of the multimodal input from the latent space representation. We train it under a denoising

Manuscript received September 10, 2017; accepted January 15, 2018. Date of publication February 2, 2018; date of current version February 22, 2018. This letter was recommended for publication by Associate Editor D. Lau and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported in part by the NSF Award IIS-1150157, in part by the NIDILRR Grant 90RE5016-01-00 via RERC TechSAge, and in part by a Google Faculty Research Award. *(Corresponding author: Daehyung Park.)*

The authors are with the Healthcare Robotics Lab, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: deric.park@gatech.edu; youna.hoshi@gmail.com; charlie.kemp@bme.gatech.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2018.2801475

autoencoding criterion [5] to prevent learning an identity function and to improve representation capability. Our LSTM-VAE-based detector detects an anomaly when the log-likelihood of the current observation given the expected distribution is lower than a threshold. We also introduce a state-based threshold to increase detection sensitivity and lower false alarms similar to [3].

We evaluated the LSTM-VAE with 352 robot-assisted feeding data collected from our previous work [4] (see Fig. 1). We newly collected a pre-training dataset from 16 able-bodied participants with 1,203 feeding executions using various food and utensils. The proposed detector was beneficial in that we could directly use high-dimensional multimodal sensory signals without significant effort for feature engineering. It can perform online anomaly detection as required to monitor task executions. In particular, it was able to set tight or loose decision boundaries depending on the variations of multimodal signals using the state-based threshold. Our method had higher area under receiver operating characteristic (ROC) curves than other baseline methods from the literature. In our evaluation, the area under the curve (AUC) was 0.044 higher than that of our previous algorithm, HMM-GP given the same data. Our new method also had a 0.064 higher AUC when we used 17-dimensional sensory signals from visual, haptic, kinematic, and auditory modalities instead of 4-dimensional hand-engineered features.

II. Related Work

Anomaly detection is known as novelty, outlier, or event detections [6]. It is also related to change-point or steady-state detections [7]. In robotics, it has been used to detect the failure of manipulation tasks [8], [9]. Researchers have often used classic approaches: support vector machine (SVM) [10], [11], self-organizing map [12], k-nearest neighbors [13], etc. To detect anomalies from time-series signals, researchers have also used HMMs [3] or Kalman filters [14].

Researchers have often fused high-dimensional input and reduced their dimension using kernel-based approaches before applying probabilistic or distance-based detections [10], [15], [16]. However, the compressed representations of outliers (i.e., anomalous data) may be inliers in latent space. Instead, we use a reconstruction-based method that attempts to reconstruct the input from its compressed representation so that it can measure reconstruction error with the anomaly score. An AE is a representative reconstruction approach that is a connected network with an encoder and a decoder [17]. It has also been applied for reconstructing time-series data using a sliding time-window [18]. However, the window method does not represent dependencies between nearby windows and a window may not contain an anomaly.

To model time-series data with its temporal dependencies, we use an LSTM network [19], which is a type of recurrent neural network (RNN). The advantages of LSTM networks over classic approaches such as window approaches or Markov chains are the representation power and the memory to track longer-term dependencies. In contrast to the HMMs, LSTM networks are able to use continuous states. Researchers have used LSTM networks for prediction in these anomaly detection domains: radio anomaly detection [20] and EEG signal anomaly detection [21]. Malhorta *et al.* introduced an LSTM-based anomaly detector (LSTM-AD) that measures the distribution of prediction errors [22]. However, the method may not predict time-series under unpredictable external changes such as manual control and load on a machine [23]. Alternatively, researchers have introduced RNN- and LSTM-based autoencoders for reconstruction-based anomaly detection [24], [25]. In particular, Malhorta *et al.* introduced an LSTM-based encoder-and-decoder (EncDec-AD) that estimates reconstruction error [23]. We also use this reconstruction scheme as a baseline method in this letter.

Another relevant approach is a variational autoencoder (VAE) [26]. Unlike an AE, a VAE models the underlying probability distribution of observations using variational inference (VI). Bayer and Osendorfer used VI to learn the underlying distribution of sequences and introduced stochastic recurrent networks [27]. Soelch *et al.* used their work to detect robot anomalies by predicting unimodal signals [28]. Bowman and Vilnis introduced an RNN-based VAE for language generation [29]. Our work also uses variational inference, but we estimate the expected distribution of input signals and a corresponding state in latent space for state-based thresholding and anomaly detection at each time step.

III. Variational Autoencoding

We review an autoencoder and a variational autoencoder. We represent a vector of multidimensional input by $\mathbf{x} \in \mathbb{R}^D$ and the corresponding latent space vector by $\mathbf{z} \in \mathcal{R}^K$, where D and K are the number of input signals and the dimension of the latent space, respectively.

A. Autoencoder (AE)

An AE is an artificial neural network that consists of sequentially connected encoder and decoder networks. It sets the target of the decoder to be equal to the input of the encoder. The encoder network learns a compressed representation (i.e., bottleneck feature or latent variable) of the input. The decoder network reconstructs the target from the compressed representation. The difference between the input and the reconstructed input is the reconstruction error. During training, the autoencoder minimizes the reconstruction error as an objective function. An AE is often used for data generation as a generative model. An AE's decoder can generate an output given an artificially assigned compressed representation.

B. Variational Autoencoder (VAE)

A VAE is a variant of an AE rooted in Bayesian inference [26]. A VAE is able to model the underlying distribution of observations $p(\mathbf{z})$ and generate new data by introducing a set of latent random variables $\mathbf{z}$. We can represent the process as $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$. However, the marginalization is computationally intractable since the search space of $\mathbf{z}$ is continuous and combinatorially large. Instead, we can represent the marginal log-likelihood of an individual point as $\log p(\mathbf{x}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathcal{L}_{\text{vae}}(\phi, \theta; \mathbf{x})$ using

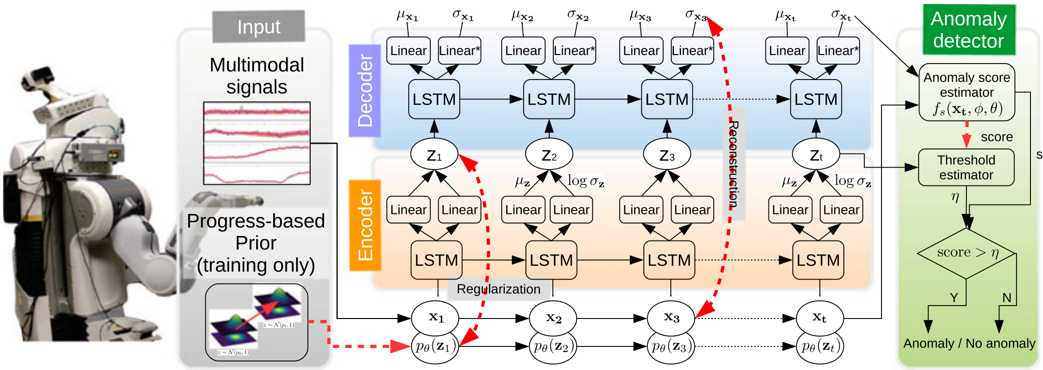

Fig. 2. Illustration of a multimodal anomaly detector with an unrolled LSTM-VAE model. The detector inputs multimodal signals to the model that compresses and reconstructs the input at each time step. The detector then reports an anomaly when a reconstruction-based anomaly score is over an estimated threshold η. In training, the model optimizes its parameters to have maximum regularization and minimum reconstruction error described in (1). The detector also trains a data-driven estimator that varies η with respect to state $\mathbf{z}$. Note that Linear* and LSTM layers have tanh and softplus activations, respectively. The red dash arrows are used for training only.

notation from [26], where D_{KL} is Kullback-Leibler divergence from a prior $p_\theta(\mathbf{z})$ to the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of $p(\mathbf{z}|\mathbf{x})$ and $\mathcal{L}_{\text{vae}}$ is the variational lower bound of the data $\mathbf{x}$ by Jensen's inequality. Note that ϕ and θ are the parameters of the encoder and the decoder, respectively.

A VAE optimizes the parameters, ϕ and θ, by maximizing the lower bound of the log likelihood, $\mathcal{L}_{\text{vae}}$,

$$\mathcal{L}_{\text{vae}} = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]. \quad (1)$$

The first term regularizes the latent variable $\mathbf{z}$ by minimizing the KL divergence between the approximated posterior and the prior of the latent variable. The second term is the reconstruction of $\mathbf{x}$ by maximizing the log-likelihood $\log p_\theta(\mathbf{x}|\mathbf{z})$ with sampling from $q_\phi(\mathbf{z}|\mathbf{x})$.

The choice of distribution types is important since a VAE models the approximated posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ from a prior $p_\theta(\mathbf{z})$ and likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. A typical choice for the posterior is a Gaussian distribution, $\mathcal{N}(\mu_{\mathbf{z}}, \Sigma_{\mathbf{z}})$, where a standard normal distribution $\mathcal{N}(0, 1)$ is used for the prior. For the likelihood, a Bernoulli distribution or multivariate Gaussian distribution is often used for binary or continuous data, respectively.

IV. LSTM-Based Variational Autoencoding

We present a long short-term memory-based variational autoencoder (LSTM-VAE). To introduce the temporal dependency of time-series data into a VAE, we combine a VAE with LSTMs by replacing the feed-forward network in a VAE to LSTMs similar to conventional temporal AEs such as an RNN Encoder-Decoder [24] and an EncDec-AD [23]. Fig. 2 shows an unrolled structure with LSTM-based encoder-and-decoder modules. Given a multimodal input $\mathbf{x}_t$ at time t, the encoder approximates the posterior $p(\mathbf{z}_t|\mathbf{x}_t)$ by feeding an LSTM's output into two linear modules to estimate the mean $\mu_{\mathbf{z}_t}$ and co-variance $\Sigma_{\mathbf{z}_t}$ of the latent variable. Then, the randomly sampled $\mathbf{z}$ from the posterior $p(\mathbf{z}_t|\mathbf{x}_t)$ feeds into the decoder's LSTM. The final outputs are the reconstruction mean $\mu_{\mathbf{x}_t}$ and co-variance $\Sigma_{\mathbf{x}_t}$.

We apply a denoising autoencoding criterion [5] to the LSTM-VAE by introducing corrupted input with Gaussian noise, $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{\text{noise}})$. We then replace the lower bound in (1) with a denoising variational lower bound $\mathcal{L}_{\text{dvae}}$ [30],

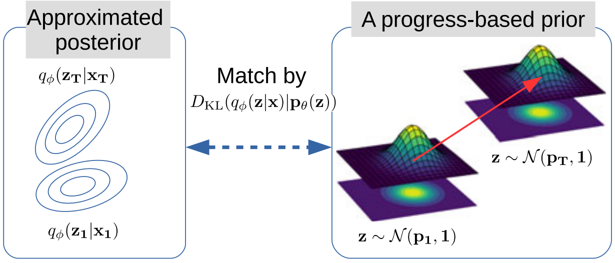
Approximated posterior
$q_\phi(\mathbf{z_T}|\mathbf{x_T})$
$q_\phi(\mathbf{z_1}|\mathbf{x_1})$
Match by
$D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})|\mathbf{p_\theta(z)})$
A progress-based prior
$\mathbf{z} \sim \mathcal{N}(\mathbf{p_T}, \mathbf{1})$
$\mathbf{z} \sim \mathcal{N}(\mathbf{p_1}, \mathbf{1})$

Fig. 3. Illustration of the progress-based prior. The center of the prior linearly changes from p_1 as initial progress to p_T as final progress.

$$\begin{aligned} \mathcal{L}_{\text{dvae}} = & -D_{KL}(\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)||p_\theta(\mathbf{z}_t)) \\ & + \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)}[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t)], \end{aligned} \quad (2)$$

where $\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)$ is an approximated posterior distribution given a corruption distribution around $\mathbf{x}_t$. Given Gaussian distributions for $p(\tilde{\mathbf{x}}|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x})$, $\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)$ can be represented as a mixture of Gaussians. For computational convenience, we use a single Gaussian, $\tilde{q}_\phi(\mathbf{z}|\mathbf{x}) \approx q_\phi(\mathbf{z}|\tilde{\mathbf{x}})$.

We introduce a progress-based prior $p(\mathbf{z}_t)$. Unlike conventional static priors using a normal distribution $\mathcal{N}(0, 1)$, we vary the center of a normal distribution as $\mathcal{N}(\mu_p, \Sigma_p)$, where μ_p and Σ_p are the center and co-variance of the underlying distribution of multimodal input, respectively (see Fig. 3). This varying prior introduces the temporal dependency of time-series data into its underlying distribution by minimizing the difference between the approximated posterior and the prior. Unlike the RNN prior of Solch *et al.* [28] and the transition prior of Karl *et al.* [31], we gradually change μ_p from p_1 to p_T as the task execution progresses. In addition, the reconstruction performance and regularization loss depend on the distribution of a selected prior. We use an isotropic normal distribution where $\Sigma_p = I$ to simplify the prior and reduce hyperparameters. Note that we have tested various priors by changing its covariance matrix, but there was no noticeable difference. We can rewrite the regularization term of $\mathcal{L}_{\text{dvae}}$ as

$$\begin{aligned} &D_{KL}(\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)||p_\theta(\mathbf{z}_t)) \\ &\approx D_{KL}(\mathcal{N}(\mu_{\mathbf{z}_t}, \Sigma_{\mathbf{z}_t})||\mathcal{N}(\mu_p, 1)). \\ &= \frac{1}{2}\left(\text{tr}(\Sigma_{\mathbf{z}_t}) + (\mu_p - \mu_{\mathbf{z}_t})^T(\mu_p - \mu_{\mathbf{z}_t}) - D - \log|\Sigma_{\mathbf{z}_t}|\right). \end{aligned}$$

To represent the distribution of high-dimensional continuous data, we use a multivariate Gaussian with a diagonal co-variance matrix. We can derive the reconstruction term in $\mathcal{L}_{\text{dvae}}$ as

$$\begin{aligned} &\mathbb{E}_{\tilde{q}_\phi(\mathbf{z}_t|\mathbf{x}_t)}[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t)] \\ &\quad = -\frac{1}{2}(\log(|\Sigma_{\mathbf{x}_t}|) + (\mathbf{x}_t - \mu_{\mathbf{x}_t})^T\Sigma_{\mathbf{x}_t}^{-1}(\mathbf{x}_t - \mu_{\mathbf{x}_t}) \\ &\qquad + D\log(2\pi)) \end{aligned} \tag{3}$$

We implemented the LSTM-VAE using stateful-LSTM models in the Keras deep learning library [32]. We trained the LSTM-VAE using an Adam optimizer with 3-dimensional latent variables and a 0.001 learning rate. We also use LSTM layers with *tanh*. Note that we are not using a sliding window in this work, but a window could be applied.

V. Anomaly Detection

We now introduce an online anomaly detection framework for multimodal sensory signals with state-based thresholding.

A. Anomaly Score

Our method detects an anomalous execution when the current anomaly score of an observation $\mathbf{x}_t$ is higher than a score threshold η,

$$\begin{cases} \text{anomaly}, & \text{if } f_s(\mathbf{x}_t, \phi, \theta) > \eta \\ \neg\text{anomaly}, & \text{otherwise}, \end{cases} \tag{4}$$

where $f_s(\mathbf{x}_t, \phi, \theta)$ is an anomaly score estimator. We define the score as the negative log-likelihood of an observation with respect to the reconstructed distribution of the observation through an encoding-decoding model,

$$f_s(\mathbf{x}_t, \phi, \theta) = -\log p(\mathbf{x}_t; \mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}), \tag{5}$$

where $\mu_{\mathbf{x}_t}$ and $\Sigma_{\mathbf{x}_t}$ are the mean and co-variance of the reconstructed distribution, $\mathcal{N}(\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t})$, from an LSTM-VAE with parameters ϕ and θ. A high score indicates an input has not been reconstructed well by the LSTM-VAE. In other words, the input has deviated greatly from the non-anomalous training data.

B. State-Based Thresholding

We introduce a varying threshold that changes over the estimated state of a task execution motivated by the dynamic threshold from our previous work [3]. Depending on the state of task executions, reconstruction quality may vary. In other words, anomaly scores in non-anomalous task executions can be high in certain states, so varying the anomaly score can reduce false alarms and improve sensitivity. In this letter, the state is the latent space representation of observations. Given a sequence of observations, the encoder of LSTM-VAE is able to compute a state at each time step. By mapping states $\mathbf{Z}$ and corresponding anomaly scores $\mathbf{S}$ from a non-anomalous dataset, our method is able to train an expected anomaly score estimator $\hat{f}_s : \mathbf{z} \to s$. We use support vector regression (SVR) to map from a multidimensional input $\mathbf{z} \in \mathbf{Z}$ to a scaler s using a radial basis function (RBF) kernel. To control sensitivity, we add a constant c into the expected score and represent the state-based threshold as $\eta = \hat{f}_s(\mathbf{z}) + c$.

Algorithm 1: Training algorithm for an LSTM-VAE-based anomaly detector.

```
input  : $\mathbf{X}_{\text{train}} \in \mathbb{R}^{N_{\text{train}} \times T \times D}$, $\mathbf{X}_{\text{val}} \in \mathbb{R}^{N_{\text{val}} \times T \times D}$
output : $\phi$, $\theta$, $f_\eta$
$\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{val}}$ = Preprocessing($\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{val}}$) ;
$\phi, \theta \leftarrow$ train LSTM-VAE with $(\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{val}})$;
$\mathbf{Z} = \emptyset, \mathbf{S} = \emptyset$ ;
for $i \leftarrow 1$ to $N_{\text{val}}$ do
    Reset the state of LSTM-VAE;
    for $j \leftarrow 1$ to $T$ do
        $\mathbf{z} \leftarrow f_\phi(\mathbf{X}_{\text{val}}(i, j))$;
        $\mu_x, \sigma_{\mathbf{x}} \leftarrow f_{\phi,\theta}(\mathbf{X}_{\text{val}}(i, j))$;
        $s \leftarrow f_s(\mathbf{x}_{\text{val}}(i, j), \mu_{\mathbf{x}}, \sigma_{\mathbf{x}})$;
        Add $\mathbf{z}$ and $s$ into $\mathbf{Z}$ and $\mathbf{S}$, respectively.
    end
end
$\hat{f}_s \leftarrow$ train an SVR with $(\mathbf{Z}, \mathbf{S})$.
```

C. Training and Testing Framework

Algorithm 1 shows the training framework of our LSTM-VAE-based anomaly detector. Given a set of non-anomalous training and validation data, $(\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{val}})$, the framework aims to output the optimized parameters (ϕ, θ) of an LSTM-VAE and an expected anomaly score estimator $\hat{f}_s$. Note that we represent N sequences of multimodal observations as $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$. N_{train} and N_{val} denote the numbers of training and validation data, respectively. We also represent the encoder and decoder functions as $f_\phi : \mathbf{x}_t \to \mathbf{z}_t$ and $g_\theta : \mathbf{z}_t \to (\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t})$, respectively. Then, we denote the function of the serially connected encoder and decoder (i.e., autoencoder) by $f_{\phi,\theta}$ with noise injection.

The framework pre-processes $\mathbf{X}_{\text{train}}$ and $\mathbf{X}_{\text{val}}$ by resampling those to have length T and normalizing their individual modalities in the range of $[0, 1]$ with respect to $\mathbf{X}_{\text{train}}$. The framework then starts to train the LSTM-VAE with respect to $\mathbf{X}_{\text{train}}$ maximizing $\mathcal{L}_{\text{dvae}}$ and stops the training when $\mathcal{L}_{\text{dvae}}$ does not increase for 4 epochs. Then it extracts a set of latent space representations and corresponding anomaly scores from $\mathbf{X}_{\text{val}}$ as the training set for $\hat{f}_s$. Finally, this framework returns the trained SVR object as well as the LSTM-VAE's parameters.

In testing, the detector aims to detect an anomaly in real time. Algorithm 2 shows the pseudo code for the online detection process. In each loop, the detector takes multimodal input $\mathbf{x}$ and scales its individual dimension with respect to the scaled $\mathbf{X}_{\text{train}}$. The detector then estimates a latent variable and the parameters

Algorithm 2: Testing algorithm for an LSTM-VAE-based anomaly detector.

```
input  : $\mathbf{x} \in \mathbb{R}^D$
output : Anomaly or $\neg$Anomaly
while True do
    $\mathbf{x} \leftarrow$ get current multimodal data;
    $\mathbf{x} \leftarrow Preprocessing(\mathbf{x})$;
    $\mathbf{z} \leftarrow f_\phi(\mathbf{x})$;
    $\mu_\mathbf{x}, \Sigma_\mathbf{x} \leftarrow f_{\phi,\theta}(\mathbf{x})$;
    if $f_s(\mathbf{x}; \mu_\mathbf{x}, \Sigma_\mathbf{x}) > \hat{f}_s(\mathbf{z}) + c$ then
        return Anomaly ;
    end
end
```

of the expected distribution. When the anomaly score of the current input is higher than η, our detector reports the current task execution is anomalous and returns the decision. We control the sensitivity of the detector by adjusting c.

VI. Experimental Setup

A. Instrumental Setup and Operations

Our system uses a PR2 from Willow Garage, a general-purpose mobile manipulator with two 7-DOF arms and powered grippers. To prevent possible hazards, we used a low-level PID controller with low gains and a 50 Hz mid-level model predictive controller from [33] without haptic feedback. We used the following sensors: an RGB-D camera with a microphone (Intel SR300) on the right wrist, a force/torque sensor (ATI Nano25) on the utensil handle, joint encoders, and current sensors. These sensors measure mouth position and sound, force on the utensil, spoon position, and joint torque, respectively.

Using a web-based graphical user interface, the user can send three commands (i.e., *scooping/stabbing*, *feeding*, and *clean spoon*) to the robot. In a typical execution, the user will send a *scooping/stabbing* command followed by a *feeding* command. The robot scoops or stabs food from a bowl given *scooping/stabbing*, and then brings the food into the user's mouth location estimated using the camera given *feeding*. The user can send *clean spoon* so that the robot can drag the spoon across a bar. The robot uses pre-defined motions which adapt to the configuration of the user and robot.

B. Data Collection

We used data from 1,555 feeding executions collected from 24 able-bodied participants. 16 participants were male and 8 were female, and the age range was 19–35. We conducted the studies with approval from the Georgia Tech Institutional Review Board (IRB).

We divided our data into two subsets: a training/testing dataset collected from our previous work [34] and a pre-training dataset. The training/testing dataset consists of data from 352 executions (160 anomalous and 192 non-anomalous) collected from 8 participants who used the feeding system with yogurt and a silicone spoon. The pre-training dataset uses data from 1,203 non-anomalous executions from 16 newly recruited participants who used various foods and utensils (see Fig. 4). The broader range of the dataset allowed us to initialize the weights of the LSTM-VAE and reduce the impact of overfitting in fine tuning. Among the dataset, 559 non-anomalous executions were from 9 participants who used 3 types of food and corresponding utensils: cottage cheese and silicone spoon, watermelon chunks and metal fork, and fruit mix and plastic spoon. An experimenter also conducted 428 non-anomalous executions as a self-study with 6 foods (yogurt, rice, fruit mix, watermelon chunks, cereal, and cottage cheese) and 5 utensils (small/large plastic spoons, a silicone spoon, and plastic/metal forks). We also collected additional data from 216 non-anomalous executions from 6 participants who used yogurt and a silicone spoon.

C. Experimental Procedure

Each participant performed anomalous and non-anomalous feeding executions while the participant, experimenters, or the system produced anomalies. We randomly determined the order of these executions. In order to approximate one form of limited mobility that people with disabilities may have, we instructed the participants to not move their upper bodies and to eat food off the utensil using their lips. We defined 12 types of representative anomalies through fault tree analysis [35]: touch by user, aggressive eating, utensil collision by user, sound from user, face occlusion, utensil miss by user, unreachable location, environmental collision, environmental noise, utensil miss by system fault, utensil collision by system fault, and system freeze (see Fig. 5). For anomalies caused by the user, we instructed the participants through demonstration videos and verbal explanation. The participant controlled the details of their actions such as timing and magnitude.

D. Pre-Processing

For each feeding execution, we collected 17 sensory signals from 5 sensors: *sound energy* (1), *force* (3) applied on the end effector, *joint torque* (7), *spoon position* (3), and *mouth position* (3), where the number in parentheses represents the dimension of signals. We zeroed the initial value and resampled each signal to have 20 Hz for the robot's actual anomaly check frequency. We then scaled signals in the non-anomalous dataset to have a value between 0 and 1. Corresponding to this scale, we also scaled signals from the anomalous dataset. Finally, we have a sequence of tuples per execution (i.e., *sequence length* × 17). Note that the executions have timing variations due to the variability of the robot's posture, each participant's seating, and human actions during the feeding.

For visualization and comparison purposes, we also extracted 4-dimensional hand-engineered features used in our previous work [4]: *sound energy*, *1st joint torque*, *accumulated force*, and *spoon-mouth distance*. Here, we used *sound energy*[1] instead of raw 44100 kHz 16 bit PCM encoding since the under sampling could miss auditory anomalies.

[1] Root mean square (RMS) of 1,024 frames.

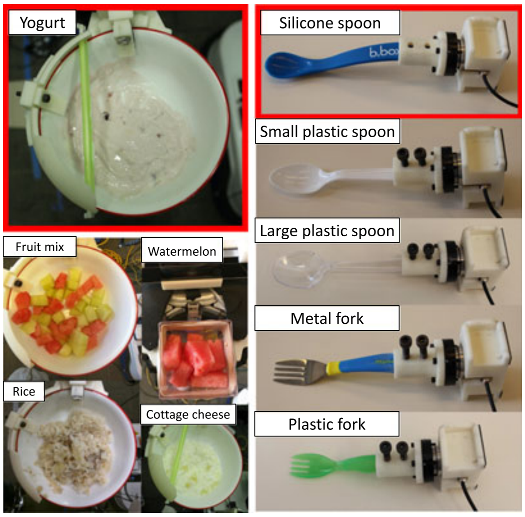

Fig. 4. **Left:** Examples of food used in our experiments. **Right:** The 3D-printed utensil handle and 5 utensils used. Red boxes show yogurt and silicone spoon used for our training/testing dataset.

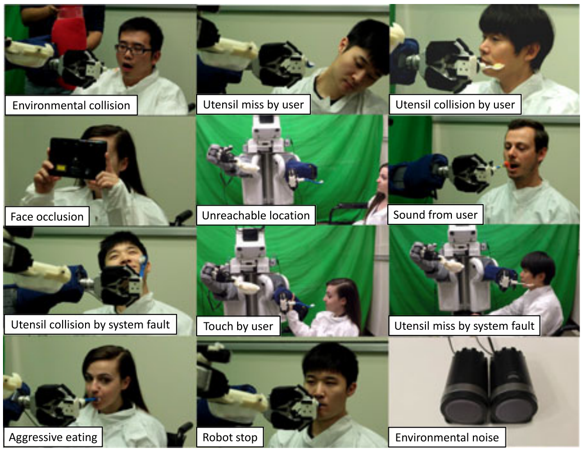

Fig. 5. 12 representative anomalies caused by the user, the environment, or the system in our experiments.

E. Baseline Methods

To evaluate the performance of the proposed method, we implemented 5-baseline methods,

- RANDOM: A random binary classifier in which we control its sensitivity by weighting a class.
- OSVM: A one-class SVM-based detector trained with only non-anomalous executions. We move a sliding window (of size 3 in time like EncDec-AD [23]) one step at a time. We control its sensitivity by adjusting the number of support vectors.
- HMM-GP: A likelihood-based classifier using an HMM introduced in [4]. We vary the likelihood threshold with respect to the distribution of hidden states.
- AE: A reconstruction-based anomaly detector using a conventional autoencoder with a 3 time-step sliding window based on [36].
- EncDec-AD: A reconstruction-based anomaly detector using an LSTM-based autoencoder [23]. We use window size $L = 3$ as in the paper, but unlike the paper we use a diagonal co-variance matrix when we model the distribution of reconstruction-error vectors.

From now on, we will also use the term LSTM-VAE to refer to our LSTM-VAE-based detector.

VII. Evaluation

We first investigated the reconstruction function of the LSTM-VAE. The upper 4 sub graphs in Fig. 6 show the expected distribution of 4 hand-engineered features from non-anomalous and anomalous feeding executions in the robot-assisted feeding task. For Fig. 6(a), the observed features (blue curves) and the mean of expected distribution (red curves) show a similar pattern of change over time. On the other hand, in anomalous executions [see Fig. 6(b) and (c)], the LSTM-VAE resulted in large deviations between observed and reconstructed *accumulated force* since the pattern by the collision is not easily observable from non-anomalous executions. Consequently, we can observe the anomaly score (blue curve) gradually increases after the onset of the deviation from the lower sub graphs. Note that the anomalous executions came from large and small face-spoon collisions caused intentionally by the user. The sound energy graphs show environmental noise only.

The anomaly score metric is effective in distinguishing anomalies. Fig. 7 shows the distributions of the anomaly scores over time of a participant's 24 anomalous and 20 non-anomalous feeding executions during leave-one-person-out cross validation. The blue and red shaded regions show the mean and standard deviation of non-anomalous and anomalous executions' anomaly scores, respectively. The score of non-anomalous executions shows a specific pattern of change with a smaller average and variance than that of anomalous executions, making anomalies easily distinguishable from non-anomalies.

The lower sub graphs of Fig. 6 also show the state-based threshold is capable of achieving a tighter anomaly decision boundary (red dash lines) than a fixed threshold over time. The expected anomaly scores (red curves) and the actual scores (blue curves) show a similar pattern of change. However, the expected score is lower than the actual score given an anomaly. Brown vertical lines show the time of anomaly detection where the first detection time matches with the initial increase of accumulated force.

We compared our LSTM-VAE with 5 other baseline methods through a leave-one-person-out cross-validation method (see Table I). Given the training/testing dataset, we used data from 7 participants for training and tested with the data from the remaining 1 participant. In this evaluation, we pre-trained each method using the pre-training dataset in addition to the dataset from the 7 participants. We then fine-tuned each method with the data from 7 participants. Note that we only trained the OSVM with the pre-training dataset and we did not succeed in training HMM-GP due to underflow errors caused by the high-dimensional input.

Our method outperformed the other methods with 0.044 higher AUC than the next best method, HMM-GP, when using 4 hand-engineered features. When using 17 sensory signals with the additional pre-training dataset, our method resulted in

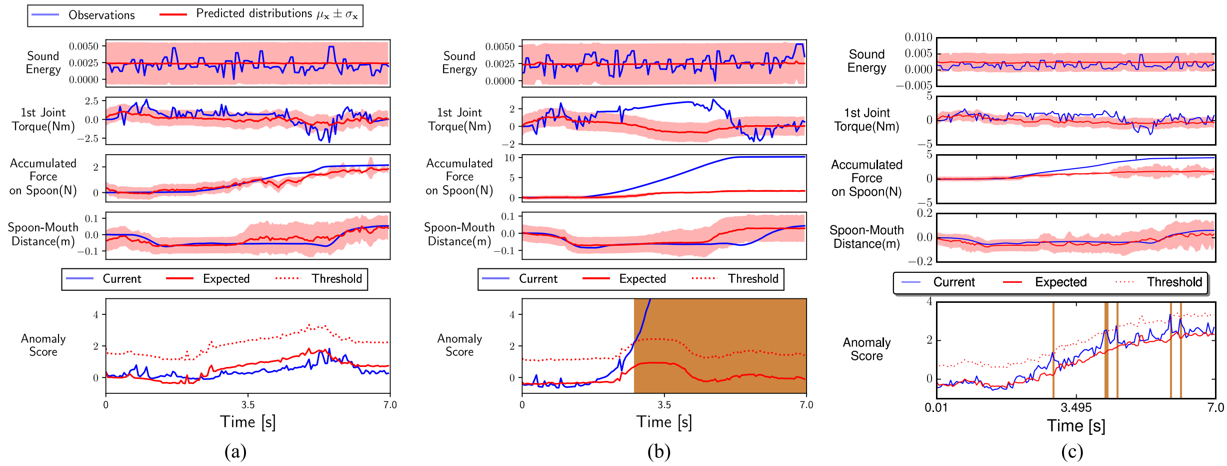

Fig. 6. Visualization of the reconstruction performance and anomaly scores over time using an LSTM-VAE. The upper four sub graphs show observations and reconstructed observations' distribution. The lower sub graphs show current and expected anomaly scores. The dashed curve shows a state-based threshold where the LSTM-VAE reports an anomaly when current anomaly score is over the threshold. Brown vertical lines represent the time of anomaly detection. (a) A non-anomalous execution. (b) An anomalous execution with large contacts. (c) An anomalous execution with small contacts.

TABLE I
COMPARISON OF THE LSTM-VAE AND 5 BASELINE METHODS WITH TWO TYPES OF INPUT SIGNALS

Input	Random	OSVM		HMM-GP	AE		EncDec-AD		LSTM-VAE
		$s = 3$	$s = 6$		$s = 3$	$s = 6$	$s = 3$	$s = 6$	
4 hand-engineered features	0.5121	0.7427	–	0.8121	0.8123	–	0.7995	–	**0.8564**
17 raw sensory signals	0.5052	0.7376	0.7408	N/A	0.8012	0.8108	0.8075	0.8021	**0.8710**

Numbers represent the area under the ROC curve (AUC). s represents the length of a window.

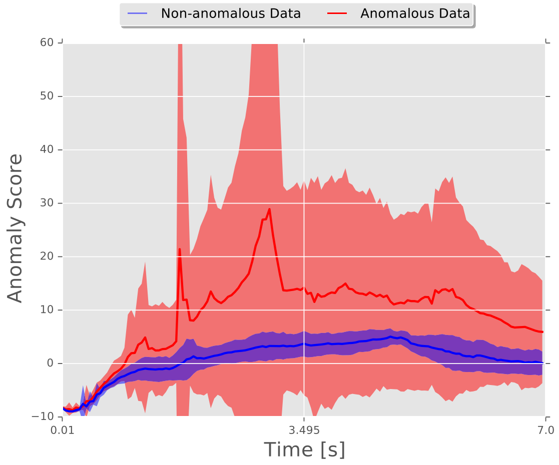

Fig. 7. Example distributions of anomaly scores from a participant's 20 non-anomalous and 24 anomalous executions over time.

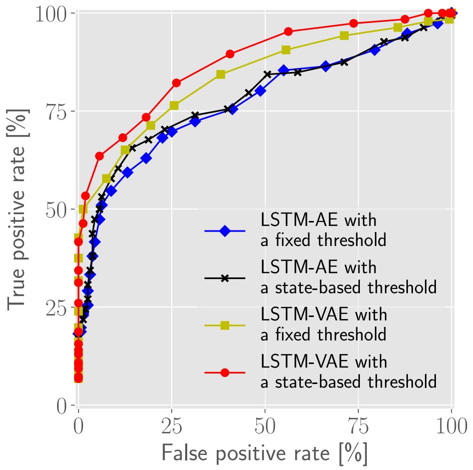

Fig. 8. Receiver operating characteristic (ROC) curves to compare the performance of LSTM-VAEs with and without a state-based threshold.

the highest performance of AUC. The time-series autoencoding methods, EncDec-AD and LSTM-VAE, improved AUC but the others did not when we increased input signals. This indicates the autoencoding is capable of extracting effective information from the high-dimensional signals without significant feature engineering effort. In addition, we tested with double the window size to investigate its effect but OSVM and AE resulted in only small improvements.

Fig. 8 shows ROC curve changes given two thresholding techniques: fixed and state-based thresholds. To investigate the influence of VAE, we implemented an LSTM-based encoder-decoder (LSTM-AE) with the two techniques by excluding VI. We used 17 sensory signals with the pre-training dataset. The LSTM-VAE with state-based thresholding outperformed that with conventional fixed thresholding, resulting in higher true positive rates given the same false positive rates. The LSTM-AE with both thresholding techniques resulted in lower true positive rates than the LSTM-VAE. Particularly, the LSTM-AE with fixed thresholding shows the lowest performance. These results indicate the VAE is helpful in reconstructing the multimodal time-series data. The results also show the VAE provides better state distribution over time for threshold regression than the vanilla AE.

VIII. Conclusion

We introduced an LSTM-VAE-based anomaly detector for multimodal anomaly detection. An LSTM-VAE models the underlying distribution of multi-dimensional signals and reconstructs the signals with expected distribution information. The detector estimated the negative log-likelihood of multimodal input with respect to the distribution as an anomaly score. By introducing a denoising autoencoding criterion and state-based thresholding, the detector successfully detected anomalies in robot-assisted feeding, resulting in higher AUC than other 5 baseline methods in the literature. Without significant effort for feature engineering, the detector with 17 raw input signals outperformed a detector trained with 4 hand-engineered features. Finally, we also showed the LSTM-VAE with the state-based decision boundary is beneficial for more sensitive anomaly detection with lower false alarms.

Acknowledgment

Dr. Kemp is a co-founder, 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

[1] D. Park, Y. K. Kim, Z. Erickson, and C. C. Kemp, "Towards assistive feeding with a general-purpose mobile manipulator," in *Proc. IEEE Int. Conf. Robot. Autom.—Workshop Human-Robot Interfaces Enhanced Phys. Interactions*, 2016.

[2] T. L. Chen *et al.*, "Robots for humanity: Using assistive robotics to empower people with disabilities," *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 30–39, 2013.

[3] D. Park, Z. Erickson, T. Bhattacharjee, and C. C. Kemp, "Multimodal execution monitoring for anomaly detection during robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom., 2016*, 2016, pp. 407–414.

[4] D. Park, H. Kim, and C. C. Kemp, "Multimodal anomaly detection for assistive robots," submitted for publication.

[5] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput. Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1541880.1541882

[7] J. Wu, Y. Chen, S. Zhou, and X. Li, "Online steady-state detection for process control using multiple change-point models and particle filters," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 688–700, Apr. 2016.

[8] A. Rodriguez, M. T. Mason, S. Srinivasa, M. Bernstein, and A. Zirbel, "Abort and retry in grasping," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 1804–1810.

[9] D. Kappler, P. Pastor, M. Kalakrishnan, M. Wuthrich, and S. Schaal, "Data-driven online decision making for autonomous manipulation," in *Proc. Robot., Sci. Syst.*, Jul. 2015.

[10] A. Rodriguez, D. Bourne, M. Mason, G. F. Rossano, and J. Wang, "Failure detection in assembly: Force signature analysis," in *Proc. IEEE Conf. Autom. Sci. Eng.*, 2010, pp. 210–215.

[11] R. Hornung, H. Urbanek, J. Klodmann, C. Osendorfer, and P. V. D. Smagt, "Model-free robot anomaly detection," in *Proc. 2014 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3676–3683.

[12] K. Häussermann, O. Zweigle, and P. Levi, "A novel framework for anomaly detection of robot behaviors," *J. Intell. Robot. Syst.*, vol. 77, no. 2, pp. 361–375, Feb. 2015.

[13] S. Ando, E. Suzuki, Y. Seki, T. Thanongphongphan, and D. Hoshino, "Ace: Anomaly clustering ensemble for multi-perspective anomaly detection in robot behaviors," in *Proc. 2011 SIAM Int. Conf. Data Mining*, 2011, pp. 1–12.

[14] J.-I. Furukawa, T. Noda, T. Teramae, and J. Morimoto, "Estimating joint movements from observed EMG signals with multiple electrodes under sensor failure situations toward safe assistive robot control," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 4985–4991.

[15] V. Sukhoy, V. Georgiev, T. Wegter, R. Sweidan, and A. Stoytchev, "Learning to slide a magnetic card through a card reader," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 2398–2404.

[16] C. Song, K. Liu, and X. Zhang, "Integration of data-level fusion model and kernel methods for degradation modeling and prognostic analysis," *IEEE Trans. Rel.*, vol. PP, no. 99, pp. 1–11, 2017.

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] K. Noda, H. Arie, Y. Suga, and T. Ogata, "Multimodal integration learning of robot behavior using deep neural networks," *Robot. Auton. Syst.*, vol. 62, no. 6, pp. 721–736, 2014.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] T. J. O'Shea, T. C. Clancy, and R. W. McGwier, "Recurrent neural radio anomaly detection," arXiv:1611.00301, 2016.

[21] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics*, 2015, pp. 1–7.

[22] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. 23rd Eur. Symp. Artifi. Neural Netw., Computat. Intell. Mach. Learn.*, 2015, p. 89.

[23] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," in *Proc. Anomaly Detection Workshop 33rd Int. Conf. Mach. Learn.*, 2016.

[24] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Language Process., ACL*, Oct. 2014, pp. 1724–1734.

[25] E. Principi, F. Vesperini, S. Squartini, and F. Piazza, "Acoustic novelty detection with adversarial autoencoders," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 3324–3330.

[26] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represen.*, Apr. 2014.

[27] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," in *Proc. NIPS 2014 Workshop Advances Variational Inf.*, 2014.

[28] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," in *Proc. ICML 2016 Anomaly Detection Workshop*, 2016.

[29] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc 20th SIGNLL Conf. Comput. Natural Language Learn.*, 2016, pp. 10–21.

[30] D. J. Im *et al.*, "Denoising criterion for variational auto-encoding framework," in *Proc. AAAI*, 2017, pp. 2059–2065.

[31] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt, "Deep variational Bayes filters: Unsupervised learning of state space models from raw data," in *Proc. Int. Conf. Learn. Represen.*, 2017.

[32] F. Chollet, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[33] A. Jain, M. D. Killpack, A. Edsinger, and C. C. Kemp, "Reaching in clutter with whole-arm tactile sensing," *The Int. J. Robot. Res.*, vol. 32, no. 4, pp. 458–482, Apr. 2013.

[34] D. Park, H. Kim, Y. Hoshi, Z. Erickson, A. Kapusta, and C. C. Kemp, "Multimodal execution monitoring for robot-assisted feeding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 5406–5413.

[35] O. Ogorodnikova, "Methodology of safety for a human robot interaction designing stage," in *Proc. 2008 Conf. Human Syst. Interactions*, 2008, pp. 452–457.

[36] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," SNU Data Mining Center, Tech. Rep., 2015.