Centaur: Robust Multimodal Fusion for Human Activity Recognition

Sanju Xaviar, Xin Yang, and Omid Ardakanian

***Abstract*— The proliferation of IoT and mobile devices equipped with heterogeneous sensors has enabled new applications that rely on the fusion of time-series emitted by sensors with different modalities. While there are promising neural network architectures for multimodal fusion, their performance falls apart quickly in the presence of consecutive missing data and noise across multiple modalities/sensors, the issues that are prevalent in real-world settings. We propose Centaur, a multimodal fusion model for human activity recognition (HAR) that is robust to these data quality issues. Centaur combines a data cleaning module, which is a denoising autoencoder with convolutional layers, and a multimodal fusion module, which is a deep convolutional neural network with the self-attention mechanism to capture cross-sensor correlation. We train Centaur using a stochastic data corruption scheme and evaluate it on five datasets that contain data generated by multiple inertial measurement units. We show that Centaur's data cleaning module outperforms two state-of-the-art autoencoder-based architectures, and its multimodal fusion module outperforms four strong baselines. Compared to two robust fusion architectures from the related work, Centaur is more robust especially to consecutive missing data that occur in multiple sensor channels, achieving 10.89–16.56% higher accuracy in the HAR task.**

***Index Terms*— Multimodal Fusion, Sensor Faults, Human Activity Recognition**

I. Introduction

THE demand for Internet of Things (IoT) and mobile devices has witnessed a steady growth in the last decade. Today, many people own smart home devices, such as security systems, smart thermostats and personal assistants, and carry multiple mobile and wearable devices, *e.g.*, they have a smartphone in their pocket and wear a smartwatch. These IoT and mobile devices are equipped with a variety of sensors. For example, smartphones and smartwatches are typically equipped with the inertial measurement unit (IMU) and global positioning system (GPS) module. The IMU combines a tri-axial accelerometer, a tri-axial gyroscope, and sometimes a tri-axial magnetometer in a system-in-package to measure specific force, angular rate, and magnetic field enabling applications such as fitness tracking [1]. Fusing multimodal data collected by sensors embedded in one or multiple such devices helps capture complementary information across different modalities [2], thereby reducing the overall uncertainty and making possible a more comprehensive understanding of human activities, health conditions, and hand gestures.

Sensor data streams are intermittent and noisy in the real world. This is primarily because sensors are used in various conditions and environments without (re)calibration and proper protection, which makes them susceptible to offsets and drifts [3], [4], in addition to dislocation, deformation, occlusion, and dirt/dust buildup [5]. For example, while the total offset and scaling error of most IMUs, including LSM9DS1 manufactured by STMicroelectronics and BNO055 by Bosch Sensortec, is within 1%, this error will be much higher if the sensor is not dynamically calibrated in the environment. Moreover, wireless sensors often send data to a node that has enough compute power to run the fusion model. For example, data generated by Vicon's Blue Trident or Xsens's MTw IMUs worn on the chest, wrist, and ankle are transferred to a smartphone or computer where they are fused for activity detection. Due to loss of connectivity or varying channel quality caused by body movements, consecutive missing data points might appear in all channels of a sensor [6]. Lastly, battery-powered sensors enter a low-energy state when the energy stored in the battery is not sufficient for their operation [7]. This could result in consecutive missing data points in some channels of a sensor until the battery is recharged.

Noise and missing data pose a challenge for the effective fusion of data from multiple sensors with different modalities. This is due to two main reasons. First, most existing deep learning techniques for multimodal fusion are not designed to handle time-varying noise and consecutive missing data [8]. Augmenting the multimodal fusion models to simultaneously address these issues and make desired inferences with high accuracy could result in complex architectures that are difficult to train and do not generalize well. Second, it is hard to capture complementary information from different sensors or modalities when data is incomplete. Rudimentary imputation methods, such as zero/mean filling and linear interpolation, may affect the cross-sensor correlation and lower the inference accuracy. As a result, most related work that considers data quality issues handles either missing data [9]–[13] or noisy measurements [14]–[16] only. To our knowledge, UniTS [17] is the only multimodal fusion model designed to be robust to consecutive missing data and noise. To train this model, the authors add Gaussian noise to training data, and randomly mask readings for simulating the effect of missing data.

Sanju Xaviar, Xin Yang and Omid Ardakanian are with the Department of Computing Science, University of Alberta, Edmonton, T6G 2R3, Canada. E-mail: xaviar@ualberta.ca, xyang18@ualberta.ca and oardakan@ualberta.ca.

In this paper, we study robust multimodal fusion for the human activity recognition (HAR) task, assuming people carry one or multiple devices, each equipped with a 9-axis or 6-axis IMU and possibly other sensors. These sensors can be heterogeneous and worn on different body parts. The fusion model should make opportunistic use of the available sensor data, handle time-varying noise as well as continuous blocks of missing data, and achieve high accuracy in the HAR task by taking advantage of the patterns that appear across multiple modalities. To satisfy these requirements, we propose a multimodal fusion model, called Centaur, that decouples data cleaning from activity recognition, such that each objective can be achieved using an effective machine learning model. Specifically, we build Centaur's data cleaning module based on a denoising autoencoder (DAE) that employs stacked convolutional layers with large kernels in the encoder to mitigate the data quality issues while extracting compressed latent representations. These representations, which are largely insensitive to missing and noisy data, are decoded using transposed convolutional layers to produce a cleaned version of the sensor data. The activity recognition module of Centaur extracts temporal feature embeddings for every sensor channel using a convolution neural network (CNN). A self-attention mechanism [18] is then applied to the feature embeddings to exploit the cross-sensor correlations for effective activity recognition. These two modules are trained independently, making it possible to attach a different inference module to the data cleaning module, *e.g.*, for gesture recognition. Our contributions are as follows:

- We propose a modular multimodal fusion model that is robust to both consecutive missing data and significant noise that varies over time. We train Centaur using a stochastic data corruption process that simulates realistic sensor faults.
- We run a microbenchmark on each module of Centaur separately, and show that the data cleaning and HAR modules both achieve outstanding performance. Specifically, the proposed convolutional DAE-based data cleaning module is compared with 2 baselines based on more complex neural network architectures, which are capable of removing noise and generating novel samples, under 4 types of sensor faults. The proposed HAR model is compared with 4 baselines that are shown to have superior performance over several multimodal fusion models in the HAR task [17].
- We conduct thorough evaluation of Centaur on five multimodal HAR datasets to study its robustness to sensor faults that might occur simultaneously. We further compare Centaur with 2 recently developed robust fusion models that can directly learn from multimodal data with sensor faults. We find that despite these faults, Centaur can effectively clean the sensor data and recognize activities, outperforming the state-of-the-art for robust fusion.

Presently, there is only a few multimodal fusion models that are robust to data quality issues that occur naturally during data capture or transmission, and even these models cannot successfully make inferences in the presence of both time-varying noise and large blocks of missing data, especially when the distributions of noise and block length are not known in advance. Centaur enables more accurate multimodal fusion for human activity recognition in the presence of these data quality issues by capturing long-term dependencies within the data using attention mechanism and complementary information from different sensors or modalities using temporal convolution layers for denoising and inference. Our code is available at `https://github.com/sustainable-computing/Centaur`

II. Related Work

A. Sensor Faults

Sensor faults and failures are common in IoT devices and sensor networks. Ni et al. [19] classify sensor faults from a data-centric perspective into outliers, spikes, stuck-at faults, and noise faults. The sensor noise is usually modeled using a time-varying multivariate Gaussian distribution, which is a convenient assumption [19]. Raposo et al. [20] classify faults into internal and external faults. Internal faults originate inside the sensor involving one or multiple physical components. External faults originate outside the sensor and include interference, environmental conditions (rain, dust, *etc.*), and overheating. Many of these issues, such as network connection, battery, and hardware issues, cause a block of successive missing data points (rather than an isolated missing data point) in one or several channels of a sensor [21].

There are several techniques to address sensor faults that cause continuous blocks of missing data across one or multiple sensor channels [22]. Assuming missing and complete data have the same distribution, Zhang et al. [23] build a sequence-to-sequence imputation model to fill in missing data sequences of varying lengths by incorporating information from earlier and later time steps. Chen et al. [13] deal with missing data by using a graph neural network (GNN) that captures modality interaction information. Tran et al. [10] impute the missing data by employing stacked residual autoencoders that model the residual between the original and predicted data. Yi et al. [24] propose a spatiotemporal multi-view-based learning method to address missing data in geo-sensory time-series. The authors integrate empirical statistical models, such as inverse distance weighting and simple exponential smoothing, with data-driven algorithms, such as user-based and item-based collaborative filtering. Statistical analysis techniques, such as mean filling, linear interpolation, and multiple imputation via chained equations (MICE) [25], are other alternatives, although they are generally worse than learning-based imputation techniques. Nevertheless, all these methods only address isolated or sequence missing data, and are not capable of denoising the existing data points.

B. Multimodal Fusion

The analysis of multimodal data, such as the data generated by mobile and wearable sensors, enables several applications, from gesture recognition [26], activity recognition [27], and gait analysis [28], [29] to autonomous driving [30] and robot-based assistant [31]. Table I shows the most commonly used

TABLE I
COMMONLY USED WEARABLE SENSORS IN DIFFERENT APPLICATIONS

Wearable sensors	Measured quantity
Inertial measurement unit	Force, angular velocity, and magnetic field
Force/pressure sensor	Force exerted on a surface
GPS module	Coordinates
RGB-D sensor	Image and depth
Electrocardiography (ECG) monitor	Heart rate
Electroencephalography (EEG) monitor	Electrical activity of the brain
Electromyography (EMG) monitor	Electrical activity produced by muscle movement and contractions
Mechanomyography (MMG) monitor	Low-frequency muscle contractions and vibrations

sensors in these applications. For instance, gait analysis techniques help study and quantify human walking patterns usually using data collected by IMU and other sensors such as heart rate monitor (producing ECG measurements) [32] and RGB-D sensor [33], [28]. This can be used to generate reference trajectories for the hip, knee, and ankle joints for bipedal robots [29]. Similarly, electromyography (EMG) and electroencephalography (EEG) monitors are often used together with IMU for the analysis of walking and postural control for rehabilitation purposes. In this work, we focus on multimodal fusion for HAR using mostly the sensors that are embedded in smartphones and smartwatches.

1) Fusion techniques: Multimodal data can be combined for classification using early, late, and multi-level fusion techniques. In early fusion [34], lower dimensional representations of multimodal data are concatenated at the input level of the application model (*e.g.,* the HAR model). A special case of early fusion is when a shared representation is learned for different modalities [1], [8]. In late fusion (*aka* decision-level fusion) [35], outputs of unimodal application models are aggregated at the end. Multi-level fusion is a hybrid approach in which fusion takes place at different stages [2].

2) Incorporating Cross-Sensor Correlation: Zhang et al. [36] learn common and modality-specific information to improve the inference capability of a multimodal emotion recognition model. The experiments were performed on audio traces and images to extract useful information. As a well-known measure of dependence and a generalized version of Pearson correlation, the Hirschfeld-Gebelein-Rényi (HGR) maximal correlation [37] is extended in [38] (named soft-HGR), to extract informative features from multiple modalities that may contain missing data.

3) Multimodal Fusion for HAR: Deep learning frameworks have been widely adopted for the classification of multimodal data, *e.g.,* for activity recognition [1], [2], [14], [39], [40]. Radu et al. [39] propose feature-concatenated and modality-specific deep neural network architectures that use DNN and CNN as base classifiers to perform activity and context recognition. DeepSense [14] integrates a CNN and an RNN to learn dependencies over time and across sensing modalities, enabling more accurate activity recognition in the presence of noise. SenseHAR [1] maps raw data collected by the available sensors to a shared low-dimensional latent space, representing a virtual sensor that is robust to the availability and variations of the sensors. The mean of these latent representations is then fed to a pre-trained HAR model to predict the activity label. DeepFusion [2] fuses readings of multiple sensors using a complex architecture. It consists of a sensor-representation (SR) module, a weighted-combination (WC) module, and a cross-sensor (CS) module. The SR module consists of multiple CNNs, one for each sensor node, to learn representations from raw heterogeneous data. The WC module uses a weighted aggregation strategy to utilize multi-sensor information. The correlation between the sensors is captured in the CS module by using a single-layer fully connected neural network, and the output vector of this module is obtained via averaging. Finally, the output of WC and CS modules are concatenated and the softmax layer is used to predict the activity label. STFNets [41] introduces a short-time Fourier neural network that can learn frequency domain representations by integrating neural networks and time-frequency analysis. A drawback of this model is its complex architecture that relies on multi-resolution layers which are computationally expensive. The multi-resolution layers consist of two-dimensional time-series data, where each dimension is transformed to the frequency domain at four different resolutions using short-time Fourier transform (STFT). Similarly, an inverse STFT operation is required to convert the data back to the time domain so that it could serve as an input to the next block. Despite significant advances made toward multimodal fusion, none of the above papers discusses how the issue of missing data or modalities can be tackled in the HAR task.

4) Robust Fusion for HAR: Some efforts have been made recently to develop a multimodal fusion model that is robust to noisy and incomplete data. SADeepSense [15] is an extension of DeepSense [14] that introduces a sensor-temporal self-attention module to take into account the reliability of heterogeneous sensors. Experiments were conducted on noise-augmented human activity and gesture recognition datasets. Yet, it does not study the effect of missing data. Nevertheless, it is possible to substitute missing data with a default value, then use SADeepSense to perform HAR, tackling both data quality issues. UniTS [17] proposes the short-time Fourier series-inspired neural network, named TS-Encoder, and employs multiple TS-Encoders to extract information in the time and frequency domains at various scales for classification tasks. Segmenting sensor data using a larger window is more favorable for UniTS as it can fully exploit the multi-scale information. Besides, the computational overhead increases as more TS-Encoders are adopted. The authors examine the robustness of the fusion model by simulating noisy environments and random missing data. For pre-processing, 45 sensor channels and windows of 256 timestamps are used to perform a 4-class human locomotion recognition on the OPPORTUNITY activity recognition dataset [42]. Our work differs from UniTS [17] in that we consider consecutive missing and noisy data simultaneously, using a stochastic corruption process described in Section III-B, while they assume each sample may be missing with a probability that is independent of the other samples, hence long periods of missing data are very unlikely in their work. Furthermore, our proposed model achieves higher classification accuracy and F1 score in the presence of these issues (see Section VII).

Novelty of this work. We have considered the most challenging scenario for robust multimodal fusion that is when multiple sensor channels exhibit noise and blocks of missing data simultaneously. This can happen due to sensor faults or communication problems. Our literature review reveals that most of the related work ignores data quality issues that occur

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3388893

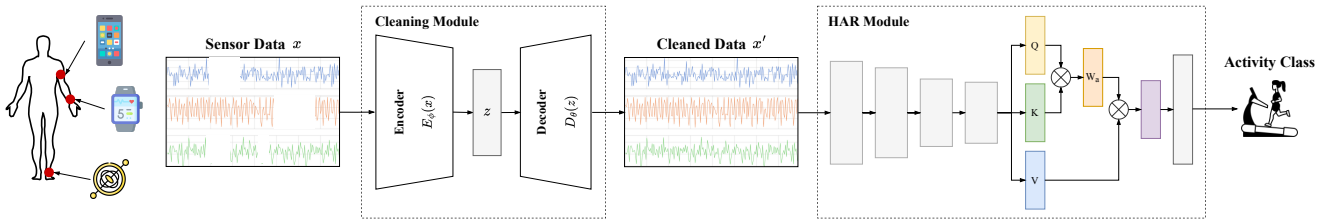

Fig. 1. Overview of Centaur's architecture after cleaning and activity recognition modules are trained independently.

naturally during data capture or transmission. The studies that consider data quality issues assume they occur in isolation, simplifying the multimodal fusion task.

III. Centaur Architecture

A. System Overview

Centaur is a robust multimodal fusion model that can perform accurate human activity recognition given intermittent and noisy data from multiple IMUs. Figure 1 depicts its architecture. Centaur takes as input a 2-dimensional matrix created by applying a sliding window to all sensor channels as described in Section VI. It has two components that are trained independently, namely a cleaning module and a human activity recognition module. The cleaning module leverages a DAE that learns an implicit mapping from corrupted to clean sensor data. The reconstructed (clean) data will be fed to a novel HAR model to perform human activity recognition.

To train the data cleaning module, we simulate sensor faults that are common in real-world settings via a stochastic corruption process described in Section III-B. We use publicly available HAR datasets that are used in prior work as ground truth, then generate corrupted sensor data by passing data segments sampled from each dataset through the corruption process. The DAE model takes as input the corrupted data and outputs a clean version by optimizing the loss, which is the distance between its output and the ground truth.[1] This decoupling of cleaning and classification tasks is advantageous because the HAR module will merely focus on extracting salient features from noise-free and complete data. Note that the HAR module is trained independently, without using the stochastic corruption process.

Once these two modules are trained, readings of heterogeneous sensors embedded in one or multiple devices are fed into the DAE model, bypassing the corruption process which is used only during training. The reconstructed (clean) data are then fed into the HAR model for activity recognition, which is a multi-class classification problem. We discuss the architecture and training of the cleaning module in Section IV and the HAR module in Section V.

B. Corruption Process

Considering the data quality issues that are present in IMU datasets (e.g., isolated and sequence missing data in PAMAP2 reported in [21]) and our past experiences with 9-axis IMUs from multiple vendors, we consider four data corruption modes that cover the following cases: sensor data in all channels are perturbed by noise; consecutive data points are missing in some channels; consecutive data points are missing in all channels of some sensors; and both noisy and missing data occur in some channels. The corruption process can be written in this form: $\tilde{x} = c_i(x, \theta_i)$, where $c_i()$ indicates the i^{th} corruption mode, x refers to the original sensor data which is assumed to be noise-free and complete, $\tilde{x}$ is the corrupted sensor data, and θ_i is the parameter(s) of the respective corruption mode.

Mode 1: Random noise is added independently to all channels and time steps: Data generated by IMUs may contain noise across all channels. This can be caused by inherent sensor noise, turn-on and in-run biases, scale factor and alignment errors [43], as well as sensor dislocation and degradation. While static calibration mitigates some of these issues, IMUs must be calibrated dynamically to keep the total error within the range specified in their data sheets (which is typically around 1%), especially when they are used in a different environment. In practice, this dynamic calibration is not performed at all times, causing the total error to greatly exceed 1%. We consider an additive white Gaussian noise, *i.e.,* the noise added to each data point (after normalization) is $n \sim N(0, \sigma)$. To control the amount of noise introduced, we change the variance of the Gaussian distribution, σ. In our experiments, we assume all sensor channels share the same σ value, but n is re-sampled from $N(0, \sigma)$ for each channel and time instant. The corruption process that generates noisy sensor data can be expressed as follows: $\tilde{x} = c_1(x, \sigma)$.

Mode 2: Consecutive missing data may appear in all channels independently: Temporary hardware issues and transitions to a low-energy state can lead to missing data over intervals of a random length. In this corruption mode, we assume such incidents may occur in all sensor channels and the length of the respective interval, l_{corr}, follows an exponential distribution, *i.e.,* $l_{corr} \sim \text{Exp}(\lambda_{corr})$. Here λ_{corr} is the rate parameter of the exponential distribution. We define the scale parameter $s_{corr} = 1/\lambda_{corr}$ to represent the corruption level; higher s_{corr} implies corruptions last longer on average. Similarly, we assume the interval in which each sensor functions normally, l_{norm}, follows an exponential distribution, $l_{norm} \sim \text{Exp}(\lambda_{norm})$ where $s_{norm} = 1/\lambda_{norm}$ is the corresponding scale parameter. For a given channel, we assume a normal interval is followed by a missing data interval and vice versa. The corruption process that generates several consecutive missing data can be expressed as follows: $\tilde{x} = c_2(x, s_{norm}, s_{corr})$. In our experiments, we fix s_{norm} and vary s_{corr} to adjust the corruption level.

Mode 3: Consecutive missing data may appear in all channels of some sensors: This type of error is common when sensors

[1]Note it is not possible to train the cleaning module using real sensor data that contains noisy and missing values because the ground truth is needed for loss calculation. Hence, we use the stochastic corruption process for training.

transmit data to a processing node via wireless connections. Varying channel quality and loss of connectivity could result in the loss of consecutive data points from all channels of some sensors. Mode 3 can be viewed as a special case of Mode 2. Hence, we assume the interval that each sensor node functions normally (or is corrupted) follows the same exponential distribution as in Mode 2. The only difference is that in Mode 2, the normal/corrupted interval is sampled independently for all channels of every sensor; whereas in Mode 3, all channels of the same sensor experience the same condition, so the normal/corrupted interval is sampled for each sensor. The corruption process of Mode 3 is expressed as: $\tilde{x} = c_3(x, s_{norm}, s_{corr})$.

Mode 4: Both noisy and missing data may appear in all channels: In the last corruption mode, we consider a challenging case where both noisy data (Mode 1) and consecutive missing data (Mode 2) can appear simultaneously in different channels. To simulate this, we first add Gaussian noise to sensor readings in all channels, then sample from $\text{Exp}(1/s_{norm})$ and $\text{Exp}(1/s_{corr})$ to determine normal and missing data intervals in each channel. This is equivalent to passing the raw data through $c_1(x, \sigma)$ and then passing the result through $c_2(x, s_{norm}, s_{corr})$. We write this corruption process as: $\tilde{x} = c_4(x, \sigma, s_{norm}, s_{corr}) = c_2(c_1(x, \sigma), s_{norm}, s_{corr})$.

IV. Convolutional Denoising Autoencoder for Data Cleaning

Centaur's cleaning module is a DAE [44] comprised of an encoder and a decoder with convolutional layers. Figure 2 shows the architecture of this DAE and the corruption process that we use to train this model. The DAE learns a mapping from the corrupted sensor readings to the cleaned sensor readings. To this end, we use a HAR dataset to train the DAE, assuming the original data are noise-free and complete. The input data is normalized to be in the range of $[0, 1]$. We then pass the original sensor data x through one mode of the corruption process as defined in Section III-B to generate the corrupted data $\tilde{x}$, which is used as input to the encoder. We study the data cleaning performance under each corruption mode separately. Note that in practice one can either perform model training using the specific corruption mode that best matches the real-world setting or use the most general corruption mode (*i.e.*, Mode 4).

The encoder in this DAE learns compressed latent representations that are insensitive to various sensor faults that could affect sensor readings. It contains four stacked two-dimensional convolutional layers (Conv2D), each followed by a ReLU activation layer to introduce non-linearity. In the first convolutional layer, we use 64 kernels that move with a stride length of 2 to extract feature representations from the corrupted data. We use a relatively large kernel size of $(5, 5)$ such that the receptive field of the convolutional kernel involves more consecutive data points in more sensor channels. This can help learn high-level patterns even when a portion of data is corrupted. In the next layers, we keep the kernel size the same, but double the number of kernels to compensate for the reduced size of the feature map caused by a large kernel. The feature map from the last convolutional layer is flattened to create a one-dimensional feature vector, which is then sent to a fully-connected dense layer to generate the latent representation z.

The decoder in this DAE reconstructs a cleaned version of the corrupted sensor data given its latent representation z. We denote the cleaned data as x'. The decoder has the same architecture as the encoder with one major difference: layers appear in reverse order. First, a fully-connected dense layer extends the latent representation z such that the extended feature vector can be reshaped and processed by a transposed two-dimensional convolutional layer (TransposedConv2D). We employ four stacked TransposedConv2D layers to recover the convolution process in the encoder. Each TransposedConv2D layer is followed by a ReLU activation layer. The number of kernels used in the decoder is the same as the corresponding Conv2D layer in the encoder, such that the output of each TransposedConv2D layer in the decoder has the same size as the input of the corresponding Conv2D layer in the encoder. Thus, the output of the last TransposedConv2D layer is the reconstructed version of the input data. To ensure the range of the reconstructed data is consistent with the normalized input data, *i.e.,* $x' \in [0, 1]$, we feed the output of the last TransposedConv2D layer to a Sigmoid function to obtain the cleaned data x'.

We use mean squared error (MSE) between the reconstructed data x' and uncorrupted data x as the loss function of the DAE. Assuming the batch size is N, the DAE loss can be expressed as:

$$\mathcal{L}_{DAE} = \frac{1}{N}\sum_{i=1}^{N}(x'_i - x_i)^2. \tag{1}$$

We use a Root Mean Squared Propagation (RMSprop) optimizer to train the model. We empirically set the learning rate to 10^{-4} with a momentum of 0.1.

V. Self-Attention CNN for HAR

We propose a deep neural network that takes advantage of convolutional and attention layers for multimodal fusion and classification of human activity. Figure 3 shows the architecture of the proposed HAR model. The model contains four stacked convolutional layers serving as a feature extractor to obtain compact embeddings of the sensor readings. A self-attention layer learns from the temporal embeddings to generate a feature map, which is flattened and fed to a fully connected layer to predict the activity.

We compose the input of our model by generating a two-dimensional data matrix, where the first dimension is the sensor time-series and the second dimension is the available sensor channels. For each convolutional layer, we use f kernels of size $(k, 1)$ to generate the convolutional feature maps, where the first dimension, k, moves along the time axis to learn and compress features in the temporal domain; the second dimension, 1, moves along the sensor channels, such that the rich information embedded in each sensor channel is retained in the convolution process and cross-channel correlation is learned later in the self-attention layer. Similar to [45], we do not employ pooling layers in our architecture. Pooling layers

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3388893

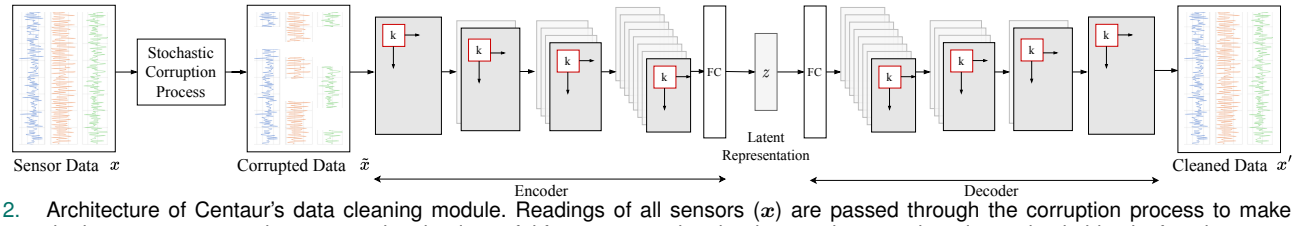

Fig. 2. Architecture of Centaur's data cleaning module. Readings of all sensors ($\boldsymbol{x}$) are passed through the corruption process to make the autoencoder learn a compressed representation that is useful for reconstructing the data, and prevent learning a simple identity function.

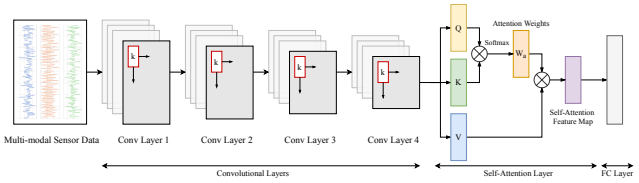

Fig. 3. Architecture of Centaur's HAR model.

are widely used in CNN-based image recognition tasks to compress the feature space. However, when processing multimodal sensor data, the number of available sensor channels or the length of sensor data segments can be very limited. Consequently, using pooling layers can significantly reduce the available information, degrading the model performance. Given the input multimodal data segment of size $(H_0 \times W_0)$, we can write the feature map size at the i^{th} convolutional layer as $(f \times H_i \times W_i)$, where $H_i = H_{i-1} - k + 1$, $W_i = W_{i-1} = W_0$, H_0 is the length of the sliding window, and W_0 is the number of available sensor channels. The four convolutional layers extract the temporal feature representation through multiple kernels for each sensor channel.[2] We reshape the output of the last convolutional layer by keeping the temporal dimension W_4, then flattening the dimension of sensor channels and the number of kernels as the input of the following attention layer.

The convolutional layers learn from the temporal feature representation and generate a multi-dimensional feature map for each sensor channel. To further exploit the correlations among different sensing modalities and sensor channels, we propose to extract such cross-channel correlations through the self-attention mechanism [18]. We transform the flattened output of the convolutional layers as the input embedding of the attention layer that has a shape of $(H_4 \times (f \times W_4))$, where H_4 is the compressed temporal dimension that determines the length of the input sequence, $f \times W_4$ is the per-channel feature representation extracted by f convolutional kernels. The query (Q), key (K), and value (V) embeddings are generated using the same input embedding to enhance the most significant cross-channel correlations via self-attention weights. By computing the scaled dot product between Q and K, and passing the results through a softmax activation function, we obtain a self-attention weight matrix W_a that determines the significance of each feature point:

$$W_a = \text{Softmax}(\frac{Q \cdot K^T}{d_k}), \tag{2}$$

where d_k is the dimension of the embedding used to scale the self-attention weights to punish large weights that would lead to very small gradients. The self-attention weight is then applied to V to generate the attention feature map. Lastly, the attention feature map is flattened and fed to a fully connected layer to predict the probability of each activity.

We use the cross-entropy loss to train the proposed human activity recognition model. A stochastic gradient descent (SGD) optimizer is used with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 10^{-4} to perform model training.

VI. Experimental Setup

A. Description of HAR Datasets

We consider a diverse set of HAR datasets that have been previously used in the activity recognition literature. Among the publicly available datasets, we select PAMAP2, OPPORTUNITY, HHAR, mHealth, and WISDM datasets for evaluation as they contain the highest number of inertial sensor modalities. Apart from IMU data, the mHealth dataset contains ECG measurements. Hence, evaluating Centaur on this dataset helps establish that it is robust to data quality issues even when the sensing modalities are more diversified. Note that in each dataset, different values are assigned to s_{norm} and s_{corr} according to the length of the time-series segments, such that we get more than a few normal and missing data intervals in each segment. Nevertheless, for all datasets, we choose the value of σ from this set $\{0.05, 0.1, 0.2, 0.3\}$. Below is a brief description of each dataset used in our evaluation.

PAMAP2 Physical Activity Monitoring Dataset [46]: This dataset consists of 12 different physical activities performed by 9 subjects wearing three 9-axis IMUs (accelerometer, gyroscope, and magnetometer). The data was sampled at 100Hz and sensor locations are as follows: 1 IMU sensor over the wrist of the dominant arm, 1 IMU on the chest, and 1 IMU on the dominant side's ankle. We follow the preprocessing steps described in [47]–[49] to segment sensor readings using a 5.12 second sliding window with 1s overlap. Then the data is down-sampled to 33.3Hz to reduce the computational overhead and normalized between 0 and 1. Each sliding window contains 171 data points after down-sampling. We set the s_{norm} for corruption Mode 2 and 3 to 80 and vary s_{corr} to be 40, 50, and 60 data points to create multiple corruption levels. 80% of the data samples are randomly chosen to train the model, and the remaining 20% are used for evaluation.

OPPORTUNITY Activity Recognition Dataset [42]: This dataset covers complex activities performed in a sensor-rich environment. We consider the on-body sensors that are mounted on the left lower arm (LLA), left upper arm (LUA), right lower arm (RLA), right upper arm (RUA), back of the torso, and feet. The dataset contains sensor readings for four

[2]We have found empirically that the HAR module performs best with 4 stacked convolutional layers, which is consistent with the observation in [45].

subjects performing daily activities while wearing sensors of different modalities. Sensor readings are collected from 5 IMUs, 12 acceleration sensors, and 2 inertial sensors installed on shoes. Each IMU has 9 channels, including a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer; each acceleration sensor contains 3 axes and each shoe sensor collects data in 16 channels, hence a total of 113 sensor channels are available. The sampling rate of the sensors is 30Hz. We consider a 5-class locomotion recognition task that involves 4 activities, namely standing, walking, sitting, lying, and a null class. We follow the preprocessing steps described in [45] (a 500ms sliding window) and use the same train-test split. We set s_{norm} of Mode 2 and 3 to 10, and s_{corr} to 4, 6, and 8 data points to create multiple corruption levels.

HHAR Heterogeneity Activity Recognition Dataset [50]: This dataset contains sensor readings from 9 users performing 6 activities: biking, sitting, standing, walking, climbing up the stairs, and climbing down the stairs. The data is collected by two types of sensors (accelerometer and gyroscope) embedded in 8 smartwatches and 4 smartphones. Each sensor produces readings in 3 dimensions, hence a total of 6 sensor channels are available per device. We only consider the data from smartphones and perform data alignment and sample uniformly separated sensor readings following the preprocessing steps described in [51]. We segment the data using 2.5-second non-overlapping windows, each window containing 100 data points. The data collected from all subjects are mixed together and we randomly draw 80% of the data for training. The remaining 20% of the data constitute the test set. We set s_{norm} of corruption Mode 2 and 3 to 50, and s_{corr} to 10, 20, and 30 data points to obtain multiple corruption levels.

mHealth Dataset [52]: This dataset contains data generated by 3 IMUs and an ECG device capturing body motion and vital signs of 10 subjects performing 12 different activities. Two Shimmer IMUs are attached to the subject's right wrist and left ankle. Each sensor contains an accelerometer, gyroscope, and magnetometer all providing 3-axis readings. The chest sensor provides 3-axis accelerometer readings and 2-channel ECG readings. Thus, 23 sensor channels are available in total, each sampled at 50Hz. We use the data for all 10 users and 12 activities but discard sensor readings when a user does perform any activity. The data are standardized and segmented using a sliding window of 40 samples with a stride length of 20 samples. We randomly select 80% of the data segments as the training set and the other segments as the test set. We set s_{norm} of corruption Mode 2 and 3 to 20, and s_{corr} to 8, 12, and 16 data points to simulate different data corruption levels.

WISDM Dataset [53]: This dataset contains sensor data obtained from 36 users performing 6 activities. The data were captured using the 3-axis accelerometer embedded in a smartphone, with a sampling rate of 20Hz. We segment sensor readings using a sliding window of 128 samples with a stride length of 20 samples. We randomly select 80% of the data segments to form the training set and the remaining data constitute the test set. We set s_{norm} of corruption Mode 2 and 3 to 70, and s_{corr} to 30, 40 and 50.

B. Baselines

We now present the baselines that are used in the next section to evaluate the two modules of Centaur.

1) Data Cleaning Baselines:

Denoising Adversarial Autoencoder (DAAE): A denoising adversarial autoencoder [54] can handle missing and noisy data. It uses both denoising and regularization to shape the latent space distribution.

In our work, we adopt the same MSE loss as introduced in Equation 1. A discriminator $D_x(z)$ is employed to match the conditional probability distribution of latent variables $q_\phi(z|\tilde{x})$ to a prior distribution $p(z)$ via adversarial training. Specifically, the latent feature sampled from the prior distribution $p(z)$ is denoted as z_{real}, the latent feature sampled from $q_\phi(z|\tilde{x})$ is denoted as z_{fake}. The discriminator aims to differentiate between z_{real} and z_{fake}, hence we express the discriminator loss as:

$$\mathcal{L}_{disc} = -\frac{1}{N}\sum_{i=0}^{N-1}\log D_x(z_{real_i}) - \frac{1}{N}\sum_{i=0}^{N-1}\log(1 - D_x(z_{fake_i})). \quad (3)$$

To ensure the latent feature can be sampled from $p(z)$, the decoder is then updated according to the prior loss:

$$\mathcal{L}_{prior} = \frac{1}{N}\sum_{i=0}^{N-1}\log(1 - D_x(z_{fake_i})). \quad (4)$$

The model training of DAAE involves three steps. First, the encoder and decoder are optimized using $\mathcal{L}_{recon}$. Then, the discriminator is optimized via $\mathcal{L}_{dis}$. Lastly, the decoder is updated via $\mathcal{L}_{prior}$ to match the prior distribution.

In our implementation of DAAE, for a fair comparison, we use the same encoder and decoder architecture as in Centaur's data cleaning module. The discriminator is a 3-layer fully connected network. DAAE is trained for 100 epochs with a batch size of 64 using the RMSProp optimizer. We set the learning rate as 10^{-4} and momentum as 0.1.

Variational Recurrent Autoencoder (VRAE): Unlike Centaur's data cleaning module that relies on convolutional layers to extract latent representations, a VRAE [55] uses recurrent neural networks that are proven effective for learning time-dependent features. Similar to a DAE, VRAE takes as input the corrupted data $\tilde{x}$ generated by one of the corruption modes and outputs the cleaned data x'. The recurrent encoder uses an LSTM layer to generate the hidden state at the current time h_t from the hidden state in the past timestamp h_{t-1} and the current sensor reading $\tilde{x}_t$. The last hidden state h_{end} compresses useful time-domain feature information learned from the entire time-series. The reparameterization trick originally introduced in [56] is used on h_{end} to sample a latent representation z from the latent space distribution Z.

The decoding process generates the clean version of the corrupted input data from the latent feature z, the probability of which can be written as $p_\theta(x'|z)$. The decoding process is similar to the encoding process, but in reverse order. First, the sampled latent feature z is decoded into the hidden state representations of the first time stamp h_1 using a fully connected layer. Then an LSTM layer performs recurrent decoding sequentially. In the t^{th} step of decoding, h_t is fed as input to recover the uncorrupted data at the first timestamp,

meanwhile generating the hidden state of the next timestamp h_{t+1}. The loss of VRAE is expressed using the evidence lower bound:

$$\mathcal{L}_{VRAE} = D_{KL}(q(z|\tilde{x})||p(z)) - \mathbb{E}_{q(z|\tilde{x})}[\log p_\theta(x'|z)]. \quad (5)$$

The first term is the KL divergence between the true posterior $q(z|\tilde{x})$ and the prior $p(z)$. The second term is the loss between the reconstructed data and the cleaned data input x'.

Mean Filling and Linear Interpolation: Under corruption Mode 2, where we only consider the existence of missing data, we further compare our data cleaning module with two widely-adopted data imputation approaches, namely mean filling and linear interpolation. Compared to DAE-based models, mean filling and linear interpolation are model-free and computationally inexpensive. The comparison helps us understand the improvement that can be made by using deep learning-based imputation techniques (although this comes at the cost of slightly increasing the computation overhead). The mean filling approach imputes the missing periods in each sensor channel using the mean value of all the normal data points in that channel. In linear interpolation, we fit a linear function between the start and end points of the missing data period, then uniformly sample the missing values according to this function.

2) Human Activity Recognition Baselines:

DeepCNN: We implement a deep convolutional neural network that contains four stacked convolutional layers. The architecture of the DeepCNN baseline is identical to the four convolutional layers in the HAR module of Centaur; thus, the comparison between DeepCNN and Centaur's HAR module demonstrates the efficacy of employing the self-attention mechanism.

DeepConvLSTM [45]: The multimodal sensor readings are processed by four convolutional layers to compress time domain information and extend the feature dimension for each sensor channel. Two LSTM-based recurrent layers extract time-dependent feature representations and pass the output through Softmax. The design of Centaur's convolutional layers is consistent with DeepConvLSTM, where pooling operations are not involved as discussed in Section V. Thus, the comparison between DeepConvLSTM and Centaur's HAR module gives insight into whether the self-attention mechanism can capture cross-sensor correlation more effectively than recurrent neural networks.

SADeepSense [15]: SADeepSense is an extension of DeepSense [14] that is designed for robust classification on multi-sensor data. SADeepSense integrates an additional Self-Attention (SA) mechanism to learn the correlation between different sensors over time. The SA module is inserted in the neural network where we combine information from the multiple sensors over time. To use SADeepSense, we first replace missing values in each channel with the arithmetic mean of normal data points in that channel. This is because SADeepSense cannot handle missing values directly.

UniTS [17]: It is a robust neural network architecture that can learn from multimodal data with artificially injected noise and dropped data points. The model consists of multiple branches of temporal-spectral encoders (TS-Encoders) extracting sensor spectrogram at different scales. In our experiments, we empirically choose the scales based on the length of the sensor data segments. In [17], UniTS is compared with state-of-the-art multimodal fusion models that were developed recently and it is shown that it outperforms these models in the HAR task.[3] Thus, we use it as a baseline for Centaur's HAR module, as well as the whole robust multimodal fusion framework.

C. Evaluation Metrics

1) Metrics for Evaluating Activity Recognition Models: We use the activity recognition accuracy and weighted F1 score averaged over 10 trials to evaluate HAR models. We compute the weighted F1 score to better represent the HAR model performance, especially when a dataset contains an uneven distribution of human activity classes. The weighted F1 score is defined as: $F1 = \sum_i \frac{w_i \cdot TP}{TP + \frac{1}{2}(FP+FN)}$, where w_i is the number of samples with activity class i over the total number of samples; TP, FP, and FN stands for true positive, false positive, and false negative, respectively.

2) Metrics for Evaluating Data Cleaning Models: We evaluate the performance of the data cleaning models using two metrics. The first metric is the root-mean-square error (RMSE) between the original (uncorrupted) sensor data and the cleaned data. An ideal data cleaning module is expected to generate cleaned sensor data with sufficiently low RMSE as it suggests that the data cleaning module is effective in reducing the distortions caused by the corruption process. Additionally, we use the performance of Centaur's HAR module (accuracy and F1 score) as our second metric. The HAR model used here is pre-trained on the original (uncorrupted) sensor data. Hence, higher HAR accuracy on the data cleaned by Centaur (or a baseline) implies more effective denoising and imputation for the target task.

3) Metrics for Evaluating End-to-end Multimodal Fusion Models: We evaluate the performance of the end-to-end sensor fusion models by measuring the accuracy and weighted F1 score obtained when using the corrupted data to perform activity recognition. In this case, we cannot use the RMSE metric because robust multimodal fusion models, such as UniTS, do not necessarily reconstruct the sensor data in its original format before they classify the activity.

D. Implementation Details

We implement Centaur, DAAE, VRAE, DeepConvLSTM, SADeepSense, and UniTS baselines using PyTorch. We followed the PyTorch implementation released by the authors of DAAE [57], UniTS [58], and an implementation of VRAE that we found online [59]. We used the Lasagne implementation provided by the authors of DeepConvLSTM as a reference [60] and made our best effort to reproduce this work using PyTorch. We used our own implementation for SADeepSense. All models are trained on an NVIDIA RTX 2080 TI GPU.

VII. Evaluation

We use microbenchmarks to investigate the efficacy of the two modules of Centaur before looking at its robustness

[3]Data quality issues are neglected in this comparison as the other models cannot handle missing and noisy data simultaneously.

TABLE II
HUMAN ACTIVITY RECOGNITION PERFORMANCE ON THE OPPORTUNITY DATASET (WITHOUT NOISE AND MISSING DATA).

	DeepCNN	DeepConvLSTM	SADeepSense	UniTS	Centaur
Accuracy	86.47	87.05	81.01	86.31	**88.78**
Weighted F1	86.31	86.91	80.23	86.27	**88.69**

and performance in comparison with our end-to-end baseline. For brevity, we only use the OPPORTUNITY dataset in our microbenchmarks, and consider all five datasets to evaluate Centaur as a whole.

A. Human Activity Recognition Module

Table II shows the performance of the proposed attention-based CNN and HAR baselines on the original (uncorrupted) OPPORTUNITY dataset[4], where the accuracy and weighted F1 score are averaged over 10 trials. The DeepCNN baseline, which uses only four stacked convolutional layers, yields average accuracy (F1 score) of 86.47% (86.31%). By incorporating recurrent dense layers, DeepConvLSTM provides a modest improvement in performance, with average accuracy (F1 score) of 87.05% (86.61%). Our attention-based model achieves the best HAR performance with average accuracy (F1 score) of 88.78% (88.69%). SADeepSense yields an average HAR accuracy (F1 score) of 81.01% (80.23%), which is lower than the other models. We attribute this to its complex architecture that makes it harder to generalize to different IMU datasets. We also evaluate the performance of UniTS on the uncorrupted OPPORTUNITY dataset and find that it achieves an accuracy of 86.31% and F1 score of 86.27%, which is slightly worse than a simple deep convolutional network. This relatively poor performance of UniTS might be due to the fact that it requires segmenting sensor data using a sufficiently large window so as to extract sensor spectrogram on multiple scales. However, in the pre-processing step, the window length of the OPPORTUNITY dataset is set to 24 samples for a fair comparison with other baselines (in particular DeepConvLSTM). This is smaller than the 512 samples used in their work [17]. To summarize, our proposed attention-based HAR module outperforms the baselines when there are no data quality issues. This implies that the proposed architecture is suitable for extracting per-channel temporal features in addition to utilizing cross-sensor information for accurate multimodal fusion.

B. Cleaning Module

We evaluate the performance of the three autoencoder-based data cleaning models (Centaur's cleaning module, DAAE, and VRAE) under the corruption modes described in Section III-B. We further compare the autoencoder-based models with mean filling and linear interpolation, which are widely adopted data imputation techniques. We assume the data corruption levels in the training and test phases are identical in these experiments. Regardless of how data is cleaned, we always pass it through the same attention-based HAR model to measure the accuracy and F1 score.

As shown in Table III, when small white Gaussian noise with $\sigma = 0.01$ is introduced, feeding the corrupted data to the HAR model directly achieves a negligible performance drop compared to when raw data is used, yielding an average accuracy (F1 score) of 89.15% (89.06%). Cleaning data with none of the data cleaning models can improve the HAR performance. When slightly increasing the noise level to $\sigma = 0.05$, the HAR performance decreases by $\sim 5\%$. Although all three autoencoder-based models improve the HAR performance by denoising the corrupted data, DAAE shows the least improvement in accuracy (by 1.10%) and weighted F1 score (by 0.54%). Examining the results for each activity, we find that DAAE improves the recognition accuracy of walking by around 16% compared to the corrupted data case, but the accuracy of standing and sitting activities drops by around 8% and 17%, respectively. Both VRAE and convolutional DAE show strong performance. That said, convolutional DAE, which is our proposed method, outperforms VRAE in terms of accuracy (F1 score) by 0.75% (0.80%). When increasing the standard deviation of the white Gaussian noise to $\sigma = 0.1$ or higher, the measurement noise significantly decreases the HAR performance (by more than 16%). DAAE is effective in denoising sensor data, yet it is still worse than VRAE and convolutional DAE. VRAE shows a more stable performance across different noise levels compared to DAAE, yet it is still worse than the proposed convolutional DAE (1–2$\%$ lower for all four noise levels). Convolutional DAE has the best denoising performance among the three autoencoder-based models. The above observations based on accuracy and F1 score are consistent with the ones that can be made by looking at the RMSE metric, where convolutional DAE shows the best denoising capability, reducing RMSE by at least $2.14\times$ when $\sigma = 0.05$ and at most $6.27\times$ when $\sigma = 0.2$.

Next we look at Mode 2 of the corruption process. As it can be seen in Table IV, the corrupted data has RMSE of 0.2479 or higher, and results in F1 score of 30% or lower (assuming missing data points are treated as zeros), even under the lowest corruption level $c_2(x, 4, 10)$. Thus, this data is unusable for activity recognition before imputation. Despite the significant impact of the consecutive missing data, we observe that all three autoencoder-based models show satisfactory performance by imputing consecutive missing data. In particular, even DAAE, which underperforms VRAE and DAE, shows an average accuracy (F1 score) of 88.05% (87.89%) under the highest corruption level. Convolutional DAE has the best performance in this case. Compared to uncorrupted data, the HAR accuracy (F1 score) only decreases by 0.53% (0.55%) for $c_2(x, 8, 10)$. Meanwhile, the RMSE is improved by $17.63\times$ compared to the corrupted data case. Although mean filling demonstrates competitive data imputation capability and has low computation overhead, the HAR accuracy on the mean-filled data is 1.28% lower than DAAE and 1.91% lower than convolutional DAE under the highest corruption level $c_2(x, 8, 10)$. We observe that linear interpolation is not an effective imputation technique and it becomes worse as the average length of the missing data interval increases. We believe this is because this imputation technique cannot recover activity-related patterns that were present in time-series data.

Looking at the results of Mode 3 in Table V, we observe very similar patterns as in Mode 2. Specifically, when using

[4]We note that the HAR task is trivial on the uncorrupted PAMAP2 dataset as Centaur's HAR model and all baselines yield an accuracy of around 99%.

TABLE III

DATA CLEANING PERFORMANCE ON OPPORTUNITY W/ MODE 1. ACCURACY (F1 SCORE) ON RAW DATA: **89.21%** (**89.12%**).

Corruption Mode 1	σ=0.01			σ=0.05			σ=0.1			σ=0.2		
	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE
Corrupted Data	89.15	89.06	0.01	84.55	84.77	0.05	67.87	68.41	0.1	44.86	42.58	0.2
DAAE	86.47	86.19	0.0300	85.65	85.31	0.0337	85.83	85.51	0.0322	83.98	83.52	0.0352
VRAE	87.89	87.73	0.0252	87.74	87.54	0.0259	87.12	86.90	0.0290	85.63	85.32	0.0343
Convolutional DAE	**88.99**	**88.88**	**0.0211**	**88.49**	**88.34**	**0.0233**	**88.12**	**87.95**	**0.0267**	**87.23**	**87.01**	**0.0319**

TABLE IV

DATA CLEANING PERFORMANCE ON OPPORTUNITY W/ MODE 2. ACCURACY (F1 SCORE) ON RAW DATA: **89.21%** (**89.12%**).

Corruption Mode 2	s_{corr}=4, s_{norm}=10			s_{corr}=6, s_{norm}=10			s_{corr}=8, s_{norm}=10		
	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE
Corrupted Data	31.58	25.24	0.2479	28.62	21.35	0.2819	27.02	19.61	0.3033
Linear Interpolation	76.92	77.40	0.0659	67.08	67.77	0.0747	60.02	60.50	0.0803
Mean Filling	88.13	87.97	0.0220	87.46	87.27	0.0346	86.77	86.56	0.0480
DAAE	88.59	88.47	0.0168	88.17	88.02	0.0201	88.05	87.89	0.0224
VRAE	88.69	88.58	0.0168	88.40	88.27	0.0199	88.30	88.16	0.0220
Convolutional DAE	**88.87**	**88.76**	**0.0130**	**88.75**	**88.64**	**0.0155**	**88.68**	**88.57**	**0.0172**

TABLE V

DATA CLEANING PERFORMANCE ON OPPORTUNITY W/ MODE 3. ACCURACY (F1 SCORE) ON RAW DATA: **89.21%** (**89.12%**).

Corruption Mode 3	s_{corr}=4, s_{norm}=10			s_{corr}=6, s_{norm}=10			s_{corr}=8, s_{norm}=10		
	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE
Corrupted Data	32.87	27.49	0.2477	29.89	23.26	0.2818	28.31	21.24	0.3034
Linear Interpolation	75.75	76.25	0.0659	66.68	67.34	0.0748	59.52	59.97	0.0804
Mean Filling	87.89	87.72	0.0219	87.06	86.87	0.0345	86.12	85.91	0.0481
DAAE	88.31	88.18	0.0185	87.49	87.31	0.0222	86.40	86.16	0.0259
VRAE	88.68	88.57	0.0149	88.45	88.33	0.0176	88.38	88.24	0.0195
Convolutional DAE	**88.91**	**88.90**	**0.0130**	**88.71**	**88.60**	**0.0156**	**88.74**	**88.62**	**0.0173**

TABLE VI

DATA CLEANING PERFORMANCE ON OPPORTUNITY W/ MODE 4. ACCURACY (F1 SCORE) ON RAW DATA: **89.21%** (**89.12%**).

Corruption Mode 4	σ=0.05, s_{corr}=4, s_{norm}=10			σ=0.1, s_{corr}=6, s_{norm}=10			σ=0.2, s_{corr}=8, s_{norm}=10		
	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE	Accuracy	F1 Score	RMSE
Corrupted Data	30.29	23.31	0.2520	28.36	21.67	0.2938	25.45	17.24	0.3448
DAAE	85.45	85.13	0.0348	82.44	81.86	0.0375	76.16	74.95	0.0447
VRAE	84.06	83.67	0.0355	81.65	81.09	0.0389	75.66	74.39	0.0446
Convolutional DAE	**88.26**	**88.10**	**0.0260**	**87.70**	**87.51**	**0.0302**	**84.74**	**84.39**	**0.0362**

the same parameters for s_{corr} and s_{norm}, directly feeding the data corrupted by Mode 3 yields slightly higher HAR accuracy and F1 score compared to Mode 2. Consequently, all three data cleaning approaches achieve a slightly better HAR accuracy and F1 score under Mode 3. Due to the similarity in the corruption level, we only discuss Mode 2 to simulate consecutive missing data in the remainder of the paper.

Last but not least, we study the data cleaning performance under the most challenging corruption mode, *i.e.,* Mode 4, where both white Gaussian noise and consecutive missing data exist. We present the result in Table VI. Convolutional DAE still shows the best performance with the lowest RMSE. When compared with the HAR accuracy (F1 score) achieved using uncorrupted data, it only reduces the performance by 0.95% (1.02%), 1.51% (1.61%), and 4.47% (4.73%) for $c_4(x, 0.05, 4, 10)$, $c_4(x, 0.1, 6, 10)$, and $c_4(x, 0.2, 8, 10)$, respectively. Looking at the performance breakdown by activity under the highest corruption level, convolutional DAE can bring the classification accuracy of walking, lying, and the null class to the same level as obtained using uncorrupted data. Compared to directly performing HAR using the corrupted data, the classification accuracy of the standing activity increases by $\sim 3.5\%$, and the accuracy of the sitting activity increases by nearly 25%. In summary, we have established that Centaur's data cleaning module has the best performance among the three autoencoder-based models.

C. End-to-end Robust Multimodal Fusion

We now compare the performance of Centaur and two robust fusion baselines, namely UniTS and SADeepSense, when they receive noisy and incomplete data. Our evaluation is carried out on the five HAR datasets described in Section VI. We consider a practical scenario in which the noise variance and average length of the missing data interval are unknown at the time the data cleaning module of Centaur is trained. Thus, for each corruption mode, we train Centaur, UniTS, and SADeepSense using a corruption process with fixed parameters, and evaluate their performance on different corruption levels. Specifically, in Mode 1 and 4, we set $\sigma = 0.1$ to train the models. As for s_{corr} and s_{norm} in Mode 2 and 4, we use the values specified for each dataset in Section VI and train the models using the middle s_{corr} value.

We first discuss the result obtained for the three corruption modes (Mode 1, 2, and 4) on the PAMAP2 dataset. Figure 4 shows the average accuracy across 5 runs and error bars show its standard deviation. When IMU data is not corrupted, the performance of UniTS and Centaur are almost on par, and about 1% higher than SADeepSense. After incorporating the

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3388893

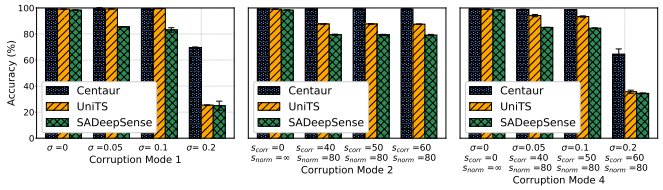

Fig. 4. Performance comparison of Centaur and baseline on PAMAP2.

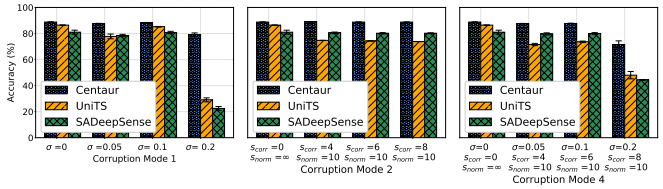

Fig. 5. Performance comparison of Centaur and baseline on OPPORTUNITY.

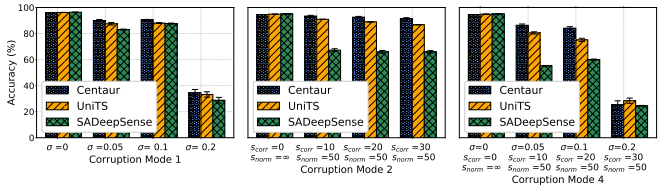

Fig. 6. Performance comparison of Centaur and baseline on HHAR.

white Gaussian noise with $\sigma = 0.05$ and $\sigma = 0.1$, the accuracy of Centaur (and UniTS) remains at the same level of $> 99\%$. However, the accuracy of SADeepSense decreases significantly to 85.53% when $\sigma = 0.05$ and to 83.11% when $\sigma = 0.1$. For the highest noise level that we considered, *i.e.*, $\sigma = 0.2$, Centaur still achieves an average accuracy (F1 score) of 69.52% (68.43%), whereas the accuracy of both UniTS and SADeepSense fall down to $\sim 25\%$. This indicates that Centaur is more robust to high, time-varying noise. Turning our attention to Mode 2, we find that all three robust fusion models show stable performance across all corruption levels, with Centaur attaining the highest HAR accuracy of 99.22%, followed by UniTS achieving an accuracy of 87.6%. SADeepSense's accuracy is around 8% lower than UniTS. However, Centaur effectively imputes the consecutive missing data and consistently shows high HAR accuracy for all noise levels. In Mode 4, Centaur achieves 98.93% accuracy for the lowest corruption level $c_4(x, 0.05, 40, 80)$, while the performance of UniTS (SADeepSense) is 4.80% (13.86%) lower than Centaur. When increasing the corruption level to $c_4(x, 0.2, 60, 80)$, we observe the HAR accuracy of UniTS and SADeepSense decrease by around 58% and 51%, respectively, whereas Centaur's accuracy only drops by 34%. Although UniTS successfully learns from data distorted by a noise process with low variance, it fails to extract useful information from different modalities in the presence of high noise and consecutive missing data. Despite the complex architecture of SADeepSense, it has the worst performance among the three robust fusion models.

Figure 5 compares the performance of Centaur and the two baselines on the OPPORTUNITY dataset. In Mode 1, Gaussian noise does not significantly affect the performance

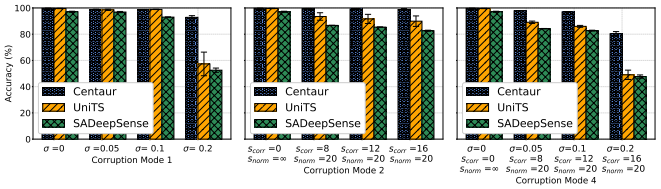

Fig. 7. Performance comparison of Centaur and baseline on mHealth.

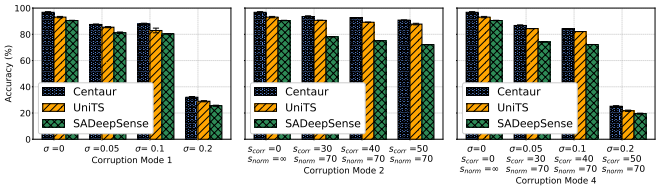

Fig. 8. Performance comparison of Centaur and baseline on WISDM.

of Centaur when σ is smaller than 0.2. UniTS maintains a high HAR accuracy when the corruption level is the same in training and testing (i.e., $\sigma = 0.1$) but its accuracy drops by 8.66% and 57.38% when the trained model is evaluated on $\sigma = 0.05$ and $\sigma = 0.2$, respectively, indicating slightly worse generalization capability compared to Centaur. Similar to the PAMAP2 results, SADeepSense performs the worst among the three robust fusion models. In Mode 2, all three models show consistent performance across all corruption levels. Centaur yields over 88% accuracy, followed by SADeepSense which yields over 80% accuracy. The accuracy of UniTS is 6% lower than SADeepSense. In Mode 4, similar to Mode 2 under the first two corruption levels, the three models show similar performance. Considering a higher corruption level, i.e., $c_4(x, 0.2, 8, 10)$, the accuracy of UniTS goes down to 48%, which is 23.57% lower than Centaur. SADeepSense's accuracy is the worst (44.34%) under the highest corruption level. Overall, we witness UniTS and SADeepSense can denoise data with a modest noise level, *i.e.*, $\sigma = 0.05$ and $\sigma = 0.1$. But, Centaur shows robust fusion capability even with higher noise and longer intervals of missing IMU data.

Next, we extend our evaluation to the HHAR dataset and present the result in Figure 6. In this case, the average accuracy of UniTS and SADeepSense are slightly higher than Centaur on uncorrupted data although the difference is not statistically significant. Specifically, the average accuracy (F1 score) for UniTS, SADeepSense, and Centaur are 96.19% (96.23%), 96.34% (96.26%), and 95.84% (95.89%), respectively. We attribute the better performance of UniTS to the larger number of TS-Encoders (with different receptive scales) that can be adopted in the HHAR dataset. However, we remark that using more TS-Encoders significantly increases the cost and complexity of training UniTS. SADeepSense performs well in this case thanks to the self-attention module that captures complex dependencies among different sensing inputs over time. Similar to the above two datasets, we find that both UniTS and Centaur perform well in Mode 2. For $c_2(x, 30, 50)$, Centaur yields an average accuracy (F1 score) of 92.74% (92.73%), outperforming UniTS by 4.80% (4.82%). However, SADeepSense's HAR accuracy is consistently lower (between $66\% - 68\%$) in all cases. In Mode 1 and Mode 4, all models

struggle to denoise the data under the highest corruption level (i.e., $c_1(x, 0.2), c_4(x, 0.2, 30, 50)$). We believe this is due to the small number of sensor channels that are present in the HHAR dataset. In fact, there are only 6 sensor channels in the HHAR dataset, whereas there is a total of 27 and 113 sensor channels in PAMAP2 and OPPORTUNITY, respectively. This restricts Centaur's ability to take advantage of cross-sensor information. Nevertheless, it still outperforms UniTS and SADeepSense with respect to accuracy and F1 score under high, time-varying noise. For example, in Mode 4, Centaur yields an average accuracy (F1 score) of 85.30% (85.29%) for $c_4(x, 0.1, 20, 50)$, outperforming UniTS by 9.15% (9.23%). SADeepSense performs poorly and only achieves around 60% accuracy.

Figure 7 compares the performance of Centaur and baseline models on the mHealth dataset. In Mode 1, both Centaur and UniTS achieve above 98.5% HAR accuracy when $\sigma = 0.05$ and $\sigma = 0.1$, outperforming SADeepSense. When the noise level increases to $\sigma = 0.2$, the HAR accuracy (F1 score) of UniTS drops to 57.33% (56.24%), whereas Centaur can still achieve an average accuracy (F1 score) of 92.85% (92.98%). In Mode 2, the performance of the three models shows a consistent pattern across the three corruption levels, where Centaur achieves the highest HAR accuracy, followed by UniTS. SADeepSense performs the worst. For the highest corruption level $c_2(x, 16, 20)$ of Mode 2, Centaur shows an average accuracy (F1 score) of 98.82% (98.83%), outperforming UniTS and SADeepSense by 8.90% (9.04%) and 16.18% (16.16%), respectively. When it comes to the most challenging case in Mode 4, i.e., $c_4(x, 0.2, 16, 20)$, Centaur can still achieve an average accuracy (F1) score of 80.35% (80.90%). However, both UniTS and SADeepSense fail to learn from the corrupted data, reaching an average HAR accuracy of 49.03% and 47.69%, respectively.

Figure 8 compares the performance of Centaur and baseline models on the WISDM dataset. In Mode 1, Gaussian noise affects the performance of all models even when σ is small. Yet, Centaur achieves the highest HAR accuracy for all σ values. In Mode 2, the performance of the three models changes only slightly and Centaur achieves the highest HAR accuracy, followed by UniTS and SADeepSense. For the highest corruption level $c_2(x, 50, 70)$ of Mode 2, Centaur shows an average accuracy (F1 score) of 90.82% (90.71%), outperforming UniTS and SADeepSense by 3.01% (2.93%) and 20.69% (20.59%), respectively. In Mode 4, Centaur yields an average accuracy (F1 score) of 84.26% (84.25%) for $c_4(x, 0.1, 40, 70)$, outperforming UniTS by 2.17% (2.2%) and SADeepSense by 12.03% (12.02%). Nevertheless, all models struggle to perform multimodal fusion under the highest corruption level (i.e. $c_4(x, 0.2, 50, 70)$). We attribute this to the small number of sensor channels that are available in the WISDM dataset (3 channels from the accelerometer).

Overall, the result presented in this section supports the claim that Centaur exhibits strong performance in the absence of noise and missing data, and is less expensive computationally and more robust to consecutive missing data and high noise than the state-of-the-art multimodal fusion models that tackle these data quality issues. When tested on datasets that have multiple sensors with different modalities, we found that Centaur captures useful cross-sensor information and takes advantage of it to improve the HAR accuracy.

VIII. Conclusion and Future Work

This paper proposes a deep neural network architecture for robust multimodal fusion. We developed a convolutional denoising autoencoder to clean noisy and incomplete sensor data, designed four corruption modes to assist with training this model, and proposed a deep convolutional neural network with the self-attention mechanism to perform human activity recognition on the data that is already cleaned. We showed that Centaur outperforms all baselines across five representative HAR datasets, and achieves high accuracy in the human activity recognition task, despite high noise and large blocks of missing data that might be the result of hardware, battery, and connectivity issues. While we reported a significant improvement over existing robust fusion models in the HAR task, our model does presently take into account the time-varying dependencies between sensors worn on different body parts. Thus, in future work, we plan to use graph neural networks and the graph attention mechanism to extract useful spatiotemporal representations that are not sensitive to data quality issues. We believe that this could further increase the performance of our model. Additionally, we will study the sensitivity of our result to the sampling rate of sensors, and explore whether we can unify the sampling rates using a smart combination of resampling and imputation.

References

[1] J. V. Jeyakumar *et al.*, "SenseHAR: A robust virtual activity sensor for smartphones and wearables," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. ACM, 2019, pp. 15–28.

[2] H. Xue *et al.*, "DeepFusion: A deep learning framework for the fusion of heterogeneous sensory data," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 151–160.

[3] S. Marathe *et al.*, "CurrentSense: A novel approach for fault and drift detection in environmental IoT sensors," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. ACM, 2021, pp. 93–105.

[4] H. Zhao and Z. Wang, "Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 943–953, 2012.

[5] A. Kashinath *et al.*, "PIRMedic: Physics-driven fault diagnosis for PIR sensors," in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. ACM, 2021, pp. 71–80.

[6] A. Vyas and S. Pal, "Optimum placement of relay nodes in wbans for improving the qos of indoor rpm system," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14 434–14 442, 2021.

[7] N. Jackson, J. Adkins, and P. Dutta, "Capacity over capacitance for reliable energy harvesting sensors," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. ACM, 2019, pp. 193–204.

[8] J. Ngiam *et al.*, "Multimodal deep learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, pp. 689–696.

[9] S. Xiang *et al.*, "Multi-source learning with block-wise missing data for alzheimer's disease prediction," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. ACM, 2013, pp. 185–193.

[10] L. Tran *et al.*, "Missing modalities imputation via cascaded residual autoencoder," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4971–4980, 2017.

[11] C. Du *et al.*, "Semi-supervised deep generative modelling of incomplete multi-modality emotional data," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 108–116.

[12] Y. H. Tsai *et al.*, "Learning factorized multimodal representations," in *International Conference on Learning Representations*, 2019.

[13] J. Chen and A. Zhang, "HGMF: Heterogeneous graph-based fusion for multimodal data with incompleteness," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1295–1305.

[14] S. Yao *et al.*, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 351–360.

[15] ——, "SADeepSense: Self-attention deep learning framework for heterogeneous on-device sensors in internet of things applications," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 1243–1251.

[16] J. Gong *et al.*, "Robust inertial motion tracking through deep sensor fusion across smart earbuds and smartphone," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 2, jun 2021.

[17] S. Li *et al.*, "UniTS: Short-time fourier inspired neural networks for sensory time series classification," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2021, pp. 234–247.

[18] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] K. Ni *et al.*, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, no. 3, jun 2009.

[20] D. Raposo *et al.*, "A taxonomy of faults for wireless sensor networks," *Journal of Network and Systems Management*, vol. 25, pp. 1–21, July 2017.

[21] A. Tawakuli, D. Kaiser, and T. Engel, "Experience: Differentiating between isolated and sequence missing data," *Journal of Data and Information Quality*, vol. 15, no. 2, Jun 2023.

[22] D. Adhikari *et al.*, "A comprehensive survey on imputation of missing data in Internet of Things," *ACM Computing Surveys*, vol. 55, no. 7, dec 2022.

[23] Y.-F. Zhang *et al.*, "SSIM—a deep learning approach for recovering missing time series sensor data," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6618–6628, 2019.

[24] X. Yi *et al.*, "ST-MVL: filling missing values in geo-sensory time series data," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.

[25] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[26] H. Gammulle *et al.*, "TMMF: Temporal multi-modal fusion for single-stage continuous gesture recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 7689–7701, 2021.

[27] F. Lin *et al.*, "Adaptive multi-modal fusion framework for activity monitoring of people with mobility disability," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4314–4324, 2022.

[28] S. K. Challa *et al.*, "An optimized-LSTM and RGB-D sensor-based human gait trajectory generator for bipedal robot walking," *IEEE Sensors Journal*, vol. 22, no. 24, pp. 24 352–24 363, 2022.

[29] V. B. Semwal *et al.*, "Gait reference trajectory generation at different walking speeds using LSTM and CNN," *Multimedia Tools and Applications*, pp. 1–19, 2023.

[30] Z. Huang *et al.*, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 781–11 790, 2020.

[31] T. Xue *et al.*, "Progress and prospects of multimodal fusion methods in physical human–robot interaction: A review," *IEEE Sensors Journal*, vol. 20, no. 18, pp. 10 355–10 370, 2020.

[32] V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," *J. Supercomput.*, vol. 77, no. 11, pp. 12 256–12 279, nov 2021.

[33] V. Bijalwan, V. B. Semwal, and T. Mandal, "Fusion of multi-sensor-based biomechanical gait analysis using vision and wearable sensor," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14 213–14 220, 2021.

[34] C. Chen *et al.*, "Learning selective sensor fusion for state estimation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[35] S. S. Saha *et al.*, *Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data*. Springer Singapore, 2021, pp. 39–53.

[36] W. Zhang *et al.*, "Multimodal emotion recognition by extracting common and modality-specific information," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 396–397.

[37] A. Rényi, "On measures of dependence," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 10, no. 3-4, pp. 441–451, 1959.

[38] L. Wang *et al.*, "An efficient approach to informative feature extraction from multimodal data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5281–5288.

[39] V. Radu *et al.*, "Multimodal deep learning for activity and context recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, jan 2018.

[40] V. Soni *et al.*, "A novel smartphone-based human activity recognition using deep learning in health care," in *Select Proceedings of 3rd International Conference on Machine Learning, Image Processing, Network Security and Data Sciences*. Springer, 2023, pp. 493–503.

[41] S. Yao *et al.*, "STFNets: Learning sensing signals from the time-frequency perspective with short-time fourier neural networks," in *The World Wide Web Conference*. ACM, 2019, pp. 2192–2202.

[42] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, 2010, pp. 233–240.

[43] "Anticipating and managing critical noise sources in mems gyroscopes. https://www.analog.com/en/technical-articles/critical-noise-sources-mems-gyroscopes.html/. [online; accessed 16-april-2021]," 2015.

[44] P. Vincent *et al.*, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[45] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[46] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *16th International Symposium on Wearable Computers*, 2012, pp. 108–109.

[47] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 1533–1540.

[48] M. Zeng *et al.*, "Understanding and improving recurrent networks for human activity recognition by continuous attention," in *Proceedings of the 2018 ACM international symposium on wearable computers*, 2018, pp. 56–63.

[49] B. Khaertdinov, E. Ghaleb, and S. Asteriadis, "Deep triplet networks with attention for sensor-based human activity recognition," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2021, pp. 1–10.

[50] A. Stisen *et al.*, "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 127–140.

[51] D. Buffelli and F. Vandin, "Attention-based deep learning framework for human activity recognition with user adaptation," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13 474–13 483, 2021.

[52] O. Banos *et al.*, "mhealthdroid: a novel framework for agile development of mobile health applications," in *Ambient Assisted Living and Daily Activities*. Springer, 2014, pp. 91–98.

[53] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, mar 2011.

[54] A. Creswell and A. A. Bharath, "Denoising adversarial autoencoders," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 968–984, 2018.

[55] O. Fabius, J. R. van Amersfoort, and D. P. Kingma, "Variational recurrent auto-encoders," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[56] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations*, 2014.

[57] "Denoising adversarial autoencoder," Available at https://github.com/ToniCreswell/pyTorch_DAAE accessed 2022.

[58] "Units," Available at https://github.com/Shuheng-Li/UniTS-Sensory-Time-Series-Classification accessed 2022.

[59] "Variational recurrent autoencoder," Available at https://github.com/tejaslodaya/timeseries-clustering-vae accessed 2022.

[60] "Deepconvlstm," Available at https://github.com/STRCWearlab/ DeepConvLSTM accessed 2022.