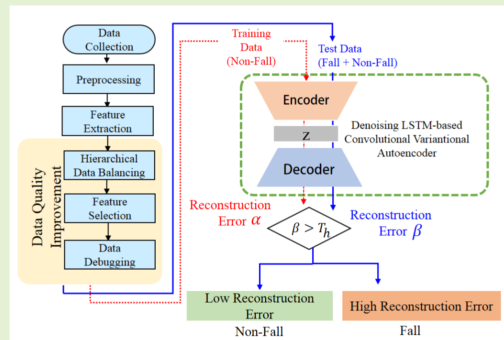
This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3388478

Fall Detection of the Elderly Using Denoising LSTM-based Convolutional Variant Autoencoder

Myung-Kyu Yi, KyungHyun Han, *Member, IEEE*, and Seong Oun Hwang, *Senior Member, IEEE*

***Abstract*—As societies age, the issue of falls has become increasingly critical for the health and safety of the elderly. Fall detection in the elderly often relies on supervised learning methods, but collecting data on falls in real situations is challenging. Additionally, the complexity of integrating deep learning models into wearable devices for real-time fall detection has been challenging due to limited computational resources. In this paper, we propose a novel fall detection method using unsupervised learning based on a denoising long short term memory (LSTM)-based convolutional variational autoencoder (CVAE) model to solve the problem of lack of fall data. By utilizing the proposed data debugging and hierarchical data balancing techniques, the proposed method achieves an F1 score of 1.0 while reducing the parameter count by 25.6 times compared to the state-of-the-art unsupervised deep learning method. The resulting model occupies only 157.65 KB of memory, making it highly suitable for integration into wearable devices.**



***Index Terms*—fall detection, unsupervised learning, variant autoencoder, wearable sensors, deep learning**

I. Introduction

LIFE expectancy worldwide has increased steadily in the past decades, leading to a larger elderly population. It is common for many individuals to live beyond the age of 60 [1], especially in developed societies. As a result, in every nation, the elderly population is growing. By the year 2030, it is anticipated that approximately one in six individuals will be over 60 [1]. In societies where the number of elderly is rising, there is growing concern about their risk of falling during daily activities. Falls can cause significant injury and even death in seniors aged 65 and above, and the death rate from such incidents is on the rise. Each year, over 14 million elderly (about one in four) report having experienced a fall [2]. While not all falls lead to serious injury, about 20% of them result in significant injuries like bone fractures or head injuries. In the year 2020, falls were responsible for more than 36,000 deaths among people aged 65 and older, making them the leading cause of injury-related deaths in this demographic [3]–[5]. For the elderly, the consequences of a fall can be grave, so it encompasses severe injuries such as fractures, the necessity for hospital care, and loss of life in the most tragic cases.

This work was supported by the National Research Foundation of Korea funded by the MSIT (Ministry of Science, ICT) under Grant 2020R1A2B5B01002145

Corresponding authors: Seong Oun Hwang (sohwang@gachon.ac.kr)

Myung-Kyu Yi, KyungHyun Han, and Seong Oun Hwang are with the Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea

To quickly detect and respond to a fall, a wearable sensor-based approach has emerged that determines whether a fall has occurred by using information collected from a device on the user's body [6]. This approach typically utilizes sensors embedded in clothing or wearable devices. These sensors are instrumental in monitoring an individual's location and movement, proving especially beneficial in detecting falls among the elderly. Accelerometers are critical in these systems, tracking physical activity and movement patterns. The integration of advanced sensors like barometers, gyroscopes, pressure sensors, infrared detectors, and motion sensors has further refined the precision of these systems. Additionally, the development of environmental sensors has led to non-intrusive detection methods. Besides wearable technology, home sensors can detect falls by analyzing motion, sound, and ground vibrations. A significant advantage of wearable-based methods is their affordability. Furthermore, these devices are easy to deploy and are user-friendly.

The core of fall detection systems is an algorithm that interprets sensor data; these algorithms initially relied on predefined thresholds for factors like acceleration [6]. Over time, Machine Learning (ML) and Deep Learning (DL) algorithms dramatically improved the classification performance between normal activities and actual falls. Traditional ML methods rely on features manually extracted from raw data and fed into classifiers like Random Forest (RF), Support Vector Machine (SVM), Decision Trees (DT), and Logistic Regression (LR) [7], [8]. This process requires specialized expertise and is limited by the need for professional knowledge.

TABLE I
PROBLEM DESCRIPTION AND THE PROPOSED FALL DETECTION APPROACH

Criterion	Existing Works	Our Approach
Method	Unsupervised learning method to address the challenge of gathering accurate data on falls	
Model	Skip-GANomaly	Denoising LSTM-based Convolutional Variant Autoencoder
Strengths	- Fusion of heart sensor with an accelerometer	- Addressing the challenge of gathering accurate data on falls - Flexibility in wearable device integration - No issue with mode collapse - Simplified model with fewer parameters
Weaknesses	- 1D to 2D conversion boosts processing demands - Complex DL models increase computational needs - Mode collapse issue - Unsuitability of model size for wearable devices	- Need to perform extra data debugging and hierarchical data balancing processing

Limitations in this method led to a research trend employing DL techniques for fall detection, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks [4], [5], [9]. Unlike traditional methods, these advanced techniques can autonomously learn and extract relevant features from raw data for classification, resulting in more reliable outcomes [10]. Fall detection accuracy can be further improved by using a hybrid model that combines various DL methods, such as CNN–gated recurrent unit (CNN–GRU), LSTM–CNN, GRU–CNN, CNN-LSTM, Convolutional-Bidirectional LSTM, and GRU (ConvBiLSTM-GRU) [11]–[13]. Moreover, DL-based fall detection technology for the elderly significantly enhances robot-assisted research by enabling robots to recognize and detect falls [14]–[18]. Robots can also learn human behavior patterns through DL algorithms and anticipate potentially dangerous situations with a high risk of falls [19]–[22].

Another way of researching fall detection using a DL model is based on vision sensors [23], in which image processing on video frames or camera images is employed. These approaches can provide high levels of accuracy in detecting falls by analyzing body posture and movement patterns. They do not require a person to wear a device, making them less intrusive. However, these methods face privacy issues, environmental constraints, and the necessity for substantial resources. Fusing the different types of sensors can enhance the effectiveness of fall detection and prevention systems [5]. A lot of work has introduced skeletal monitoring to detect falls by the elderly [24]. Despite the remarkable achievements of previous work on fall detection systems, several challenges still remain.

1) Obtaining fall data is a significant challenge because falls are unpredictable by nature [10]. Additionally, publicly available datasets on falls are limited [10], [25].
2) An unsupervised learning approach can be an excellent solution for overcoming the problem of collecting fall data. However, more parameters and complex DL model structures are needed to provide high accuracy, making it challenging to deploy them in wearable devices with limited memory and computational resources [26]. For example, the STM32 Nucleo-144 for wearable devices only supports 1 MB of flash memory [27]. Developing DL models that maintain high accuracy with fewer parameters is essential for implementing fall detection systems in wearable devices, which is a very complex challenge [28], [29].

Nho et al. [30] proposed a novel approach for detecting falls using a Generative Adversarial Network (GAN) integrated with a fusion of a heart sensor and an accelerometer to enhance performance. The approach, called UI-GAN, has shown remarkable results in fall detection. However, the approach has some limitations, as shown in Table I. One primary concern is converting one-dimensional data to a two-dimensional format, which is both time-consuming and computationally intensive, thus increasing the processing time and resource needs [31]. Furthermore, the Skip-GANomaly model [32] used in UI-GAN is complex and relies on a large amount of data for practical training and inference, making it a challenge to use in low-resource environments. The conversion to 2D also risks losing or distorting crucial aspects of the original data, potentially affecting the model's accuracy and efficacy. Finally, the Skip-GANomaly model, as a GAN framework component, is susceptible to mode collapse [33], further complicating the training of GANs.

To tackle the challenges of implementing efficient fall detection in resource-constrained environments such as wearable devices, we introduce an unsupervised learning-based fall detection method using a denoising LSTM-based CVAE model. We chose the CVAE model because it provides a stable training process and objective assessment of falls by quantifying data distributions. The proposed method can achieve high accuracy with a significantly reduced parameter count due to courtesy of innovative data debugging and hierarchical data balancing techniques. The proposed method guarantees stable learning and is meticulously optimized for wearable devices' constrained computational and memory capacities, ensuring both efficiency and practicality in application. Our contributions are as follows.

1) To the best of our knowledge, this paper is the first to employ an unsupervised learning method using the denoising LSTM-based CVAE model. The proposed method for detecting falls significantly reduces the need to collect specific fall-related data, as it uses training data representing normal daily activities. The training data can be easily collected naturally, making the approach more practical and applicable in real-world scenarios.
2) Through careful analysis and deep insight into the dataset, we can achieve high accuracy by improving the quality and balancing of training data, even when utilizing a simple model. By applying these observations to hierarchical data balancing and data debug-

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.2024.3388478

ging techniques, we improve fall detection accuracy by covering the learning ability of complex models with many parameters. As a result, it reduces the number of parameters by about 25.6 times, compared to the approach presented in [30], while still achieving an F1 score of 1.0 (occupying only 157.65 KB of memory), making it ideal for integration into wearable devices.

Our implementation is in the public domain and available at https://github.com/kainos14/CVAE. The rest of this paper is organized as follows. Preliminaries for this paper are introduced in Section II. Section III describes the details of the proposed fall detection method. Section IV presents and analyzes the experimental results. Section V concludes the paper.

II. Preliminaries

This section covers the preliminary information required to understand the proposed fall detection method. We explain the limitations of the existing art created for fall detection data. Then, we describe the VAE model used to apply our proposed fall detection method.

A. The Limitations of the Existing Art

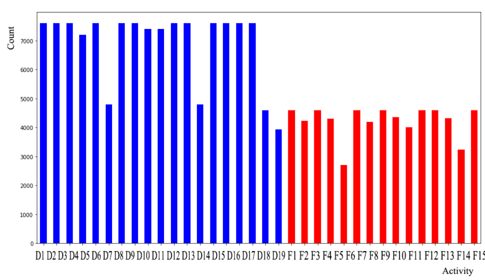


Fig. 1. Data distribution for each activity in the SisFall dataset

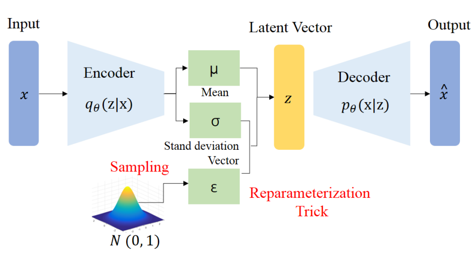


Fig. 2. The Variational AutoEncoder (VAE) architecture

First, a significant challenge in fall detection is the difficulty in obtaining fall data [6], [10], [34], which raises the following problems: privacy issues related to sensitive personal information, accurately capturing real-time data owing to its unpredictable nature, ethical considerations regarding the elderly, and technical limitations. [6], [10].

Secondly, a significant challenge in fall detection research is reliance on datasets created in laboratory settings [10], [34], which often lack diversity in participants and scenarios. These datasets typically include a limited number of subjects, and capture only a narrow range of falls, failing to represent the complexity of real-life situations. Additionally, since the data from simulated falls is created by volunteers in controlled environments like gyms and laboratories, doubts can be raised about the effectiveness of these methods in real-world contexts, especially among older adults. While non-fall data, i.e., Activities of Daily Living (ADL), can easily be gathered in realistic environments, falling is rare, making compiling a comprehensive and representative dataset difficult.

Thirdly, fall detection systems should be predominantly implemented with wearable or edge devices due to their need for continuous monitoring, which is crucial in detecting and responding to emergencies like falls. However, their limited memory and computational resources can make it challenging to deploy them in wearable or edge devices like field-programmable gate arrays (FPGAs). For example, the STM32 Nucleo-144 in a wearable device has a flash memory capacity of 1 MB. Recent Xilinx Virtex UltraScale+ FPGAs have a maximum of 500 MB of on-chip memory [35]. Still, the EmotionNet model [36] for low-cost embedded devices, which uses five convolutional layers, requires memory usage of about 761 MB [37]. Therefore, it is essential to develop efficient DL models with fewer parameters while keeping accuracy high, particularly for practical applications such as wearable or edge devices [28], [29].

Finally, a crucial problem with fall detection datasets is data balancing [25], [34]. Imbalance in the data is inevitable because actual falls occur less frequently than ADLs, which can lead to false positives (non-falls identified as falls) or false negatives (actual falls that are not detected). Strategies to lessen this imbalance include under-sampling and over-sampling techniques. It is well known that data balancing can mitigate bias issues, enabling the model to learn more equitably across all classes, potentially leading to improved overall accuracy [25]. However, data on falls and ADLs are also composed of various sub-data, so there is an imbalance within them, and previous work did not address this. For example, the SisFall dataset [38] shown in Fig. 1 has 15 distinct types of fall, such as falling forward, backward, or laterally from different postures (red bars) and 19 ADLs, such as jogging, sitting, climbing stairs, descending stairs, and walking at different speeds (blue bars). As shown in Fig. 1, we found differences in the total amount of data between falls and ADLs and differences in the amount within falls (or ADLs). Therefore, based on this observation, we built a more accurate and reliable fall detection system with a balanced dataset.

B. Variational Autoencoders

This section describes an overview of the VAE model applied to our proposed fall detection method. VAEs are generative models specifically designed to capture the underlying probability distribution of a given dataset to produce different samples [40]. As shown in Fig. 2, the VAE consists of an encoder, $q_\theta(z|x)$, and a decoder, $p_\theta(x|z)$. The goal of an encoder is to capture the essence of the input data, x, and

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.2024.3388478

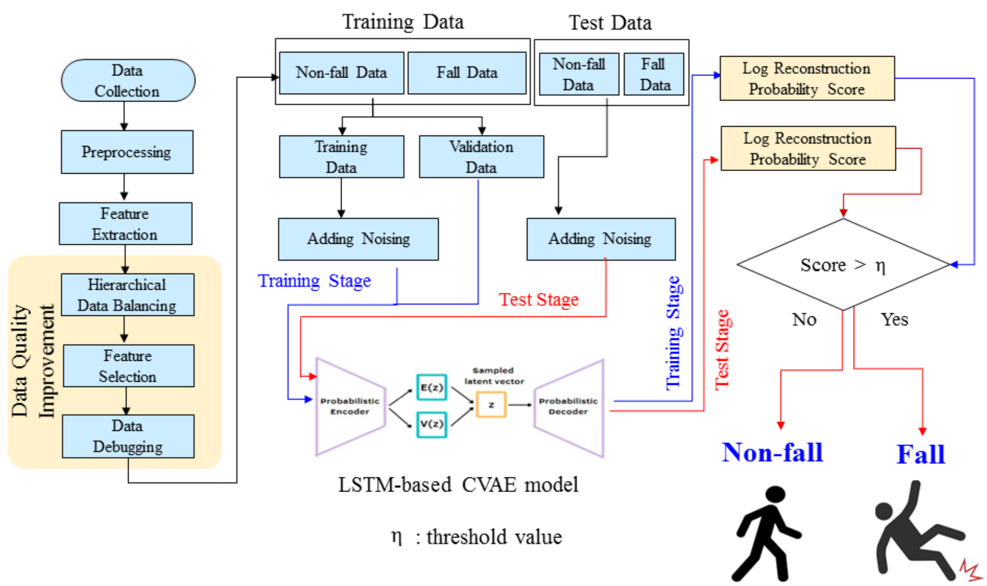


Fig. 3. Flowchart of the proposed fall detection method

express it in latent space, z, by modeling the conditional probability, $p(z|x)$, known as the posterior distribution. The decoder in a VAE takes latent representation z and returns reconstruction x. The variational lower bound of the input data's marginal likelihood serves as the VAE's objective function, because the marginal likelihood is challenging to manage. Obtaining the marginal likelihood involves adding up the marginal likelihoods of several data points, as seen in Eq. (1):

$$\log p_\theta\left(x_1, x_2, \ldots, x_N\right)=\sum_{i=1}^N \log p_\theta\left(x_i\right) \tag{1}$$

VAEs employ a distinct approach called variational inference to estimate the likelihood of various data points. This method approximates posterior distribution $q_\theta(z|x)$. Instead of direct output of latent variable z, the encoder in a VAE is designed to produce the parameters of the distribution, specifically the mean (μ) and standard deviation (σ) of z, through iterative training on input data. Rather than obtaining z straight from the encoder, it is derived from a stochastic process. This process generates a random sampling parameter, ϵ, from a standard normal distribution. Latent variable z is then computed by resampling, using this random component combined with the learned μ and σ. This approach allows the model to capture the probabilistic nature of the data representation in the latent space in Eq. (2):

$$z=\mu(x)+\sigma(x)*\varepsilon, \quad \varepsilon \sim N(0,1) \tag{2}$$

The loss of the VAE for a given data point x is represented in Eq. (3):

$$\mathcal{L}(\theta, \phi; x)=-\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]+D_{KL}(q_\phi(z|x)\|p(z)) \tag{3}$$

In Eq. (3), θ and ϕ are the parameters of the decoder and encoder networks, respectively. The first term is the reconstruction loss. It measures how effectively the decoder reconstructs original data x from latent variable z. The second term is Kullback-Leibler (KL) divergence. It measures how much the encoder's distribution, $q_\phi(z|x)$, over latent variable z diverges from prior distribution $p(z)$, typically assumed to be a standard normal distribution. This loss function is designed to achieve a dual purpose: it seeks to reconstruct the original data precisely, and simultaneously ensures the properties of the latent space are desirable. It is typically achieved through the KL divergence term, which promotes a normal distribution in the latent variables.

III. The Proposed Fall Detection Method

Fig. 3 presents a flowchart of the proposed fall detection method using the CVAE model developed to detect falls suffered by the elderly. During training, the denoising LSTM-based CVAE model learns to represent and reconstruct typical ADL patterns efficiently. When test data, including fall and ADL input, are added to the denoising LSTM-based CVAE model, it attempts to reconstruct the input based on patterns it learned during training. In this process, normal data as ADLs, similar to the training data, are typically reconstructed with high fidelity, demonstrating low reconstruction errors. However, in the case of abnormal data, which include fall events not represented in the training dataset, the model often struggles to reconstruct the input accurately, resulting in significantly higher reconstruction errors. This distinction in reconstruction error forms the basis of anomaly detection. 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.2024.3388478

TABLE II
THE LIST OF SELECTED FEATURES FOR THE PROPOSED METHOD

	Feature
1	Mean
2	Median
3	Mean absolute deviation
4	Standard deviation
5	Kurtosis
6	Skewness
7	Sum magnitude area
8	Angle between z-axis and vertical
9	Slope value
10	Yaw angles
11	Pitch angles
12	Roll angles

model can differentiate between normal and abnormal data by setting a specific threshold for the reconstruction error [39].

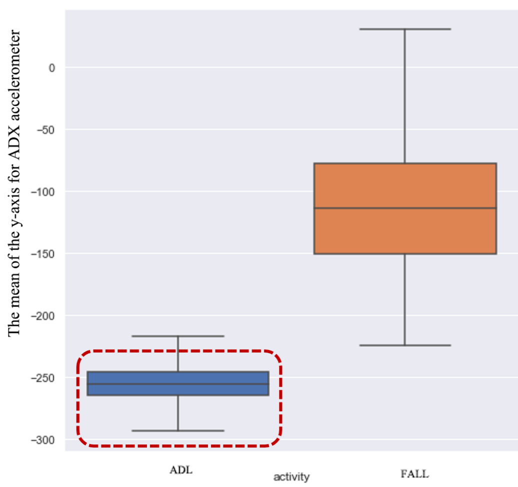


Fig. 4. Example of the proposed data debugging technique on the SisFall dataset

The preprocessing phase involves several fundamental procedures, including eliminating outliers and missing values, data cleansing, transformation, and scaling. At this stage, noise reduction techniques such as Gaussian filtering and transformations in time and frequency domains may be employed to enhance the granularity of activity recognition. Although these techniques are interesting issues in research for fall detection, they fall outside the scope of this paper.

A. Feature Extraction

The crucial step of feature extraction follows the initial preprocessing phase. In this stage, we calculate statistical feature vectors, such as the mean and standard deviation, from each segmented data window. Feature extraction focuses on highlighting the most significant attributes of the data. This process is essential for enhancing the performance of advanced algorithms in both learning and predictive tasks. It is also a commonly employed technique in various fall detection systems [6]. Wearable devices are often constrained by limited processing power and storage capacity. Therefore, minimizing

Algorithm 1: Hierarchical Data Balancing

Input: Set of ADLs $A_n = \{A_1, A_2, ..., A_n\}$
  Set of Falls $F_m = \{F_1, F_2, ..., F_m\}$
Output: Balanced set $A'_n = \{A'_1, A'_2, ..., A'_n\}$
  Balanced set $F'_m = \{F'_1, F'_2, ..., F'_m\}$
// Initialization
$A_h \leftarrow \min(A_n)$, $F_h \leftarrow \min(F_m)$
$A'_n \leftarrow \varnothing$, $F'_m \leftarrow \varnothing$
Δ : Ratio of ADLs to Falls
if $\left(\Delta \times \sum_{i=1}^{n} A_i > \sum_{i=1}^{m} F_i\right)$ **then**
  // More ADLs data than Falls data
  $F_h \leftarrow (\Delta \times \sum_{i=1}^{n} A_i)/m$
  // Set threshold to balance ADLs and Falls
else
  // More Falls data than ADLs data
  $A_h \leftarrow \sum_{i=1}^{m} F_i/(\Delta \times n)$
end
for $i \leftarrow 1$ **to** n **do**
  $\sigma = \mid A_h - A_i \mid$
  if $\sigma \neq 0$ **then**
    Adjust A_i by σ samples and save to A'_i
    $A'_n = A'_n \bigcup \{A'_i\}$
    // Adjust the amount of ADL data
  end
end
for $i \leftarrow 1$ **to** m **do**
  $\delta = \mid F_h - F_i \mid$
  if $\delta \neq 0$ **then**
    Adjust F_i by δ samples and save to F'_i
    $F'_m = F'_m \bigcup \{F'_i\}$
    // Adjust the amount of Falls data
  end
end
return (A'_n, F'_m)

Algorithm 2: Feature Selection

Input: Set $f_n = \{f_1, f_2, ..., f_n\}$
Output: Set $f_m = \{f_1, f_2, ..., f_m\}$
//Initialization
  $T_h \leftarrow Initialvalue$
for $i \leftarrow 1, n$ **do**
  $correlation\_matrix[i] = f_i.corr()$
end
$m = 0$;
for $i \leftarrow 1, n$ **do**
  $correlation\_matrix[i] = f_i.corr()$
  if $\mid correlation\_matrix[i] \mid >= T_h)$ **then**
    $f_m \leftarrow f_i$
    m++
  end
end
return (f_m)

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.2024.3388478

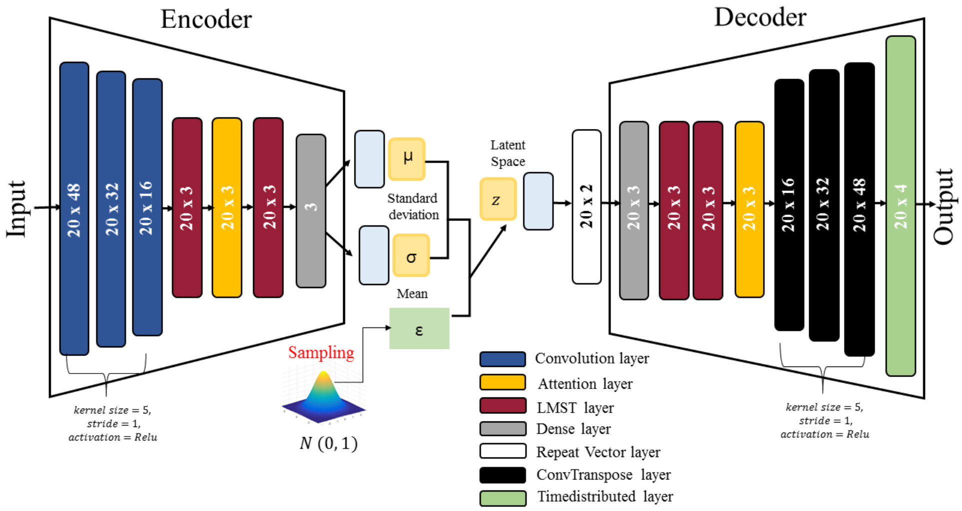


Fig. 5. The proposed denosing LSTM-based CVAE model architecture

Algorithm 3: Data Debugging

Input: Set $f_m = \{f_1, f_2, ..., f_n\} \leftarrow$ *Algorithm* 2
Output: X_{train}
$f_{max} = \emptyset$;
for $i \leftarrow 1, m$ **do**
  Calculate the correlation between the label and f_i
  Save to $correlation\_matrix[i]$
  if $f_{max} <| correlation\_matrix[i] |$ **then**
    $f_{max} \leftarrow | correlation\_matrix[i] |$
  end
end
$ADL \leftarrow f_{max}[label] ==$ 'ADL'
  // Select only rows with the label 'ADL'
$FALL \leftarrow f_{max}[label] ==$ 'FALL'
  // Select only rows with the label 'FALL'
Exploring data ranges in ADL and FALL
$X_{train} \leftarrow$ Exclude data overlapping with FALL in ADL
return (X_{train})

the computational demands to align with wearable device requirements is crucial. The proposed method sets priorities by using a minimal number of features to achieve a more efficient and lightweight model. Based on previous work on fall detection, we select only 12 features from accelerometer and gyroscope data for the proposed method. Table II lists the features selected for the proposed fall detection method.

B. Hierarchical Data Balancing

Data imbalance is a significant issue in datasets, particularly in fall detection, where datasets often disproportionately represent various activities of daily living (ADLs) and falls. Traditional methods have primarily focused on balancing the overall distribution of falls and ADLs without considering the nuances of sub-activities within these categories. The proposed hierarchical data balancing approach introduces an innovative solution by addressing the imbalance between falls and ADLs, and fine-tuning sub-activities distribution within each category. This method leverages over-sampling and under-sampling techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE) and random sampling [25], to achieve a more granular balance. As shown in Algorithm 1, ADLs and falls are denoted $A_n = A_1, A_2, ..., A_n$ and $F_m = F_1, F_2, ..., F_m$, respectively. Initially, the minimum number of ADLs required for training, A_{min}, is assigned to A_h, and the minimum number of falls, f_{min}, is assigned to f_h. Considering Δ, the ratio of ADLs to falls, we set threshold values A_h and F_h for ADLs and falls, respectively. These thresholds ensure that ADLs and falls satisfy a given amount of data. Then, the algorithm adjusts the amount of data for each activity within the falls and ADLs by generating or reducing data to match the established threshold values for A_h and F_h.

C. Feature Selection and Data Debugging

The proposed feature selection and debugging aims to improve fall detection accuracy by improving data quality. To this end, we search for the distribution of the most correlated features in the data and then remove the training data within the range overlapping the fall data of the corresponding feature. In Algorithm 2, upon initial input of feature values, the algorithm computes the correlation between the labels and each feature, storing this in a correlation matrix. Only features exhibiting a correlation value exceeding threshold T_h are selected from this matrix. It is well known that there is a correlation when the correlation value is more significant than 0.3 [41]. In this paper, we use a default value of 0.4. This feature selection process optimizes the predictive model for fall detection by excluding independent variables that are not significantly related to the dependent variable.

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.2024.3388478

As shown in Algorithm 3, data debugging is performed using the features selected in Algorithm 2 as input. From among them, we choose the feature with the highest correlation value to be f_{max}. Then, f_{max} is again split into falls and ADLs (non-falls) according to their labels to compare the respective range values. Any ADL data overlapping the fall data range is excluded from this process. In general, falls and ADLs can be distinguished by the distribution of feature values with high correlation values in most of the fall detection datasets [34]. After removing data with a range overlapping the fall range from ADL, the accuracy for fall detection can be improved by only training the model with the refined data. Fig. 4 shows an example of the proposed data debugging technique. In the SisFall dataset, the mean value on the y-axis of ADX acceleration has the most largest correlation value (i.e., f_{max}). So, the proposed data debugging algorithm selects ADL data for training, excising any data overlapping with a range overlapping range of fall data (denoted by the red dotted line in Fig. 4). Features with high correlation clearly distinguish between falls and ADLs, so their overlapping area is small [26], [34], and accuracy can be improved by only using the precise ADL data.

D. The Denoising LSTM-based CVAE Model

1) The Model architecture: To the best of our knowledge, this paper is the first to apply the denoising LSTM-based CVAE model to detect falls. The proposed architecture of the denoising LSTM-based CVAE model is shown in Fig. 5. The blue, red, green, grey, white, yellow, and black boxes represent convolutional, LSTM, TimeDistributed, dense, repeat vector, attention, and ConvTranspose layers, respectively. We use convolutional layers in the baseline VAE model because they can extract spatial features from each frame or time point. In contrast, LSTM layers can analyze the temporal evolution of these features. In Fig. 5, the numbers in the boxes indicate the shape of the layer. The proposed LSTM-based CVAE model consists of an eight-layer encoder that includes the input layer and a nine-layer decoder that includes the output layer. We set each layer's number and the hyperparameter value for CVAE by using trial and error [39].

2) Attention Mechanism: We add an attention layer to a CVAE model to focus on significant features and improve interpretability. The attention mechanism was first introduced by Bahdanau et al. [42] in natural language processing to efficiently manage large volumes of high-dimensional information in neural networks. Selectively focusing on critical data, and filtering out less relevant details, enhances the network's training efficiency by minimizing information overload. At each time step t, the attention module computes a new vector, c_t, obtained as a weighted sum of all hidden states from the current and previous time steps, as follows:

$$c_t = \sum_{i=1}^{T} \alpha_{t,i} h_i \tag{4}$$

Vector c_t is computed as a weighted sum of hidden states $h_1, ..., h_{T_x}$. Each hidden state h_i is weighted by $\alpha_{t,i}$.

The set of $\alpha_{t,i}$ is the set of weights for how much each source hidden state should attend to each output state. Attention weight $\alpha_{t,i}$ for each hidden state h_i is computed by applying a softmax operation as follows:

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{j=1}^{T} exp(e_{t,j})} \tag{5}$$

The alignment scores, $e_{t,i}$, are calculated using the hidden state produced in the previous time step.

3) The Denoising Variational Autoencoder: We build the Denoising CVAE (DCVAE) model, which is an advanced CVAE model explicitly designed to handle and reconstruct data compromised by noise. What sets the DCVAE apart from the standard CVAE is its unique training approach: it is trained on datasets where the input is deliberately corrupted with noise, typically using a Gaussian distribution. This training enables the DCVAE to process noisy input effectively and produce clean output, showcasing its ability to discern and preserve the essential features of the underlying distribution of the input data. Mean squared error (MSE) is used to train the DCVAE to reconstruct the original data from the corrupted data, as seen in Eq. (6):

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{6}$$

In Eq. (6), n is the total number of data points, y_i is the individual element of the original data, and $\hat{y}_i$ represents the corresponding element in the reconstructed data. We utilize a sample dataset including falls and ADLs to determine the optimal threshold [39], [43]. This threshold demonstrates precise identification of fall detection within this sample dataset and is then successfully applied to the entire test set. We set the threshold for successful fall detection below 1.0 for testing loss [43].

IV. Performance Evaluation

A. Evaluation Metrics and Datasets

This section describes the experimental results and analysis of the fall detection method. The evaluation metrics used in benchmarking our method are given in Eqs. (7)-(10).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

Table III describes datasets and sensors used for an experiment. The MobiFall dataset [44] contains data from 11 volunteers, comprising six males (22–32 years old, 1.69–1.89 m tall, 64–102 kg) and five females (22–36 years old, 1.60–1.72 m tall, 50–90 kg). Nine participants performed the falls and ADLs, while two performed only falls. ADLs were chosen based on their commonness and similarity to actual falls,

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.2024.3388478

TABLE III
DESCRIPTION OF DATASETS AND SENSORS USED FOR AN EXPERIMENT

Dataset	Number of Types (ADLs/Falls)	Number of Subjects (Female/Male)	Ages(Years)	Positions of the Sensing Points	Type of Sensor	Sensors	Year
MobiFall	9/4	24 (7/17)	22∼ 47	Thigh (Trouser Pocket)	1 smartphone	Accelerometer, Gyroscope, Orientation	2013
MobiAct	12/4	57 (42/15)	20 ∼ 47	Thigh (Trouser Pocket)	1 smartphone	Accelerometer, Gyroscope, Orientation	2016
SisFall	19/15	38 (19/19)	19 ∼ 75	Waist	1 external sensing mote (self-developed prototype)	Accelerometer, Gyroscope	2017
FallAllD	44/34	15 (8/7)	21 ∼ 53	Waist, Wrist, Neck	Three identical data-loggers (developed by RF-Track)	Accelerometer, Gyroscope, Magnetometer, Barometer	2020

which may produce false positives. Data were collected using a Samsung Galaxy S3 equipped with an LSM330DLC inertial module, which includes a 3D accelerometer and gyroscope. The accelerometer was recorded at an average frequency of 87 Hz with a standard deviation of about 7.6 ms in the sampling period. In comparison, the gyroscope and orientation data were captured at an average rate of 200 Hz, with a standard deviation of approximately 0.3 ms in the sampling period.

The MobiAct dataset [46] is a publicly accessible dataset acquired from a Samsung Galaxy S3 smartphone with the LSM330DLC inertial module (3D accelerometer and gyroscope). It contains four types of falls, 12 actions in daily living, and a scenario of daily living actions by 66 subjects from more than 3200 trials. Fifty-nine subjects performed nine of the eleven ADLs, 19 performed all the ADLs, and 19 subjects performed the five sequences. The subjects' ages ranged between 20 and 47 years, the height ranged from 160 cm to 193 cm, and the weight varied from 50 kg to 120 kg. The average profile of the subject that occurs based on the described characteristics is 26 years old, 176 cm in height, and 76 kg in weight.

The SisFall dataset [38] is an extensively utilized open dataset featuring falls and ADLs captured using a custom-built device that includes two types of accelerometers and a gyroscope. This dataset contains 19 different ADLs and 15 types of falls executed by 23 young adults. It also consists of 15 kinds of ADL performed by 14 healthy, independent individuals over age 62, along with data from a 60-year-old participant who performed all the ADLs and all kinds of falls. The participants with elderly females (62-75 years) and males (60-71 years) vary in height from 1.50 to 1.71 meters and weight from 50 to 102 kilograms. The participants with adult females and males, both 19-30 years old, range in height from 1.49 to 1.83 meters and in weight from 42 to 81 kilograms. The dataset was recorded with a self-developed embedded device composed of a Kinets MKL25Z128VLK4 microcontroller, an analog devices ADXL345 accelerometer, a Freescale MMA8451Q accelerometer, an ITG3200 gyroscope, an SD card for recording, and a 1000 mA/h generic battery.

FallAllD [9] is a public dataset documenting simulated human falls and everyday actions by 15 participants (eight male and seven female participants, aged 21 to 53, weighing 48 to 85 kg, and ranging in height from 158 to 187 cm), tracked with data loggers on their waist, wrist, and neck. It captures motion through accelerometers, gyroscopes, magnetometers, and barometers using LSM9DS and MS5607-02BA03 devices developed by RF-Track. Notice that the proposed method uses only accelerometer and gyroscope data to ease of application to wearable devices. For all datasets, five-fold cross-validation was applied. It enabled five distinct train-validate cycles, with 80% of training and validation data and 20% for testing. Of the 80%, 10% was allocated for validation.

TABLE IV
PERFORMANCE EVALUATION FOR THE PROPOSED FALL DETECTION METHOD

Dataset	MobiFall	MobiAct	SisFall	FallAllD
Accuracy Score	1.0000	0.9942	0.9689	0.9342
Precision	1.0000	0.9911	0.9669	0.9761
Recall	1.0000	0.9977	0.9850	0.9196
F1	1.0000	0.9944	0.9759	0.9470

B. Experimental Results

Table IV shows performance evaluations of the proposed fall detection method on the MobiAct, MobiFall, SisFall, and FallAllD datasets. With the MobiAct dataset, the proposed model achieved an F1 score of 0.9944. Using the MobiFall dataset resulted in an outstanding performance with an F1 score of 1.0000. The proposed model achieved an F1 score of 0.9759 on the SisFall dataset and an F1 score of 0.9470 on the FallAllD dataset, indicating a slightly lower performance than the other datasets.

Overall, using the four datasets demonstrated high performance. Based on the confusion matrices of the proposed fall detection method, as shown in Fig. 6, the receiver-operating characteristic (ROC) curves are presented along with the computed areas under the curve (AUC) for ADLs and falls. Fig. 6 shows that the AUCs for the ADLs and falls were 1.0, 1.0, 0.96, and 0.94, for the MobiAct, MobiFall, SisFall and FallAllD datasets, respectively. The best performance was obtained from the proposed fall detection method, with a notable decline in the true positive rate concurrent with a reduction in the false positive rate on the MobiFall and MobiAct datasets.

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.2024.3388478

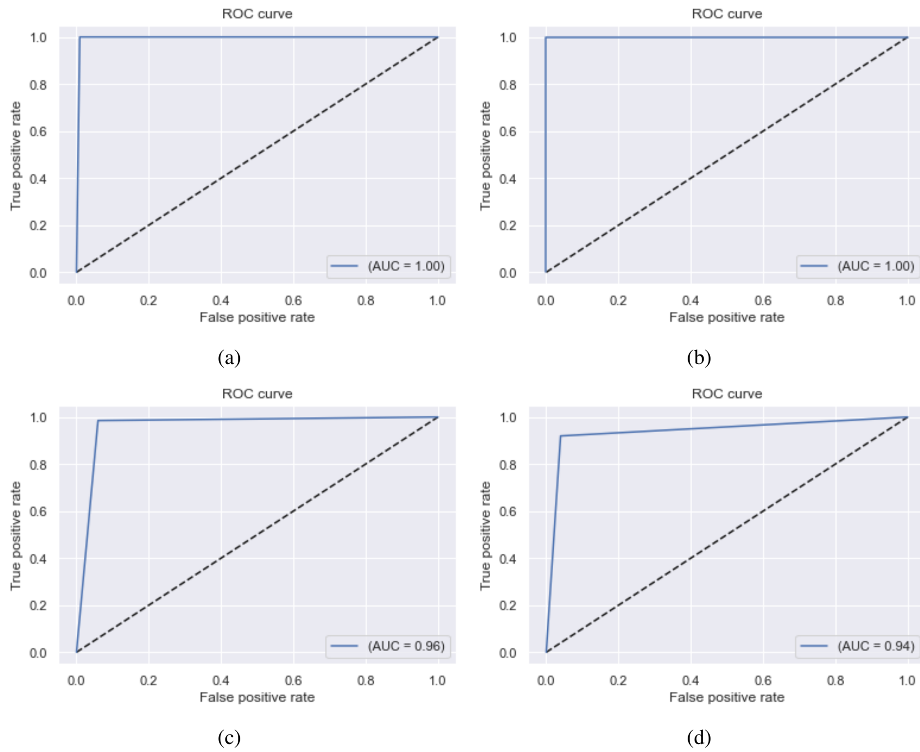


Fig. 6. ROC curves obtained from the proposed fall detection method by using (a) the MobiAct dataset, (b) the MobiFall dataset, (c) the SisFall dataset, and (d) the FallALLD dataset

TABLE V
OVERALL PERFORMANCE OF THE PROPOSED FALL DETECTION METHOD

	Learning Types	MobiFall	MobiAct	SisFall	FallALLD	Model	Number of parameters
[45]	Supervised	0.9985	-		-	DBN	-
[47]	Supervised	0.9103	-	-	-	Gradient Boosting	-
[48]	Supervised	-	0.9682	-	-	GRU	38.669
[49]	Supervised	-	0.9897	-	-	Ensemble Methods	-
[50]	Supervised	-	0.9890	-	-	TCN-GRU	-
[51]	Supervised	-	-	0.9940	-	CNN	78,482
[52]	Supervised	-	-	0.9898	-	CNN-LSTM	-
[53]	Supervised	-	-	0.8968	-	CNN-LSTM	-
[54]	Supervised	-	-	-	0.9432	CNN-GRU	-
[30]	Unsupervised	1.0000	0.9941		-	Skip-GANomaly	Discriminator: 164,545 Generator: 870,273
This work	Unsupervised	**1.0000**	**0.9953**	**0.9759**	**0.9470**	CVAE	Encoder & Decoder: 40,359

Table V shows the overall performance of the proposed fall detection method compared to previously reported numerous results that used supervised and unsupervised fall detection methods. The existing fall detection method using a DL model based on a supervised technique showed the best F1 score of 0.9985 [45], 0.9897 [49], 0.9898 [52], and 0.9432 [54] on the MobiFall, the MobiAct, the SisFall, and FallAllD datasets, respectively. The supervised fall detection methods show similar performance to the proposed method. However, the supervised learning fall detection methods require as much fall data as possible for training to achieve high accuracy. With only training data for ADLs, the proposed fall detection method achieved an F1 score of 1.0, 1,0, 0.9759, and 0.9470 when using the MobiFall, MobiAct, SisFall, and FallAllD datasets, respectively. In contrast to previous work [54], the proposed methods achieved high accuracy by only employing accelerometers and gyroscopes, eliminating the need for additional sensors. For the unsupervised learning methods, we can see the total number of parameters for the Skip-GANomaly model used in [30] and the proposed DCVAE model, as shown in Table IV. The proposed DCVAE model has 25.6× fewer network parameters than the approach presented in [30], and the accuracy of the DL model using only the accelerometer improved slightly. Having more parameters adds to the complexity of the DL model and requires more data and computational resources for training. Also, the inference process becomes more memory-intensive and time-consuming, potentially hindering the development of fall detection systems and diminishing their practicality. The proposed fall detection method utilized significantly fewer parameters (resulting in a memory size of only 157.65 KB), achieved the highest F1 score with all four datasets compared to existing works. It

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.2024.3388478

is highly suitable for implementation in resource-constrained environments, such as wearable devices, to detect elderly falls effectively.

C. Ablation Study

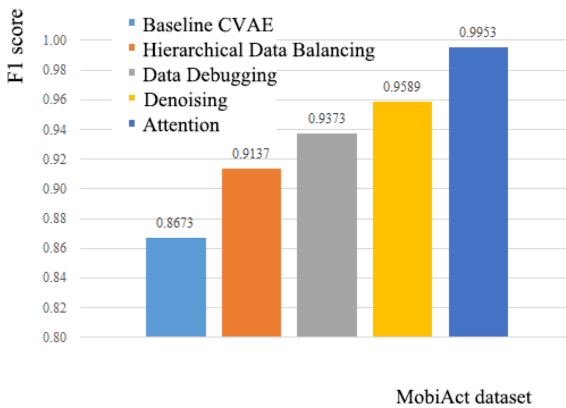


Fig. 7. The ablation study of the proposed fall detection method when using the MobiAct dataset

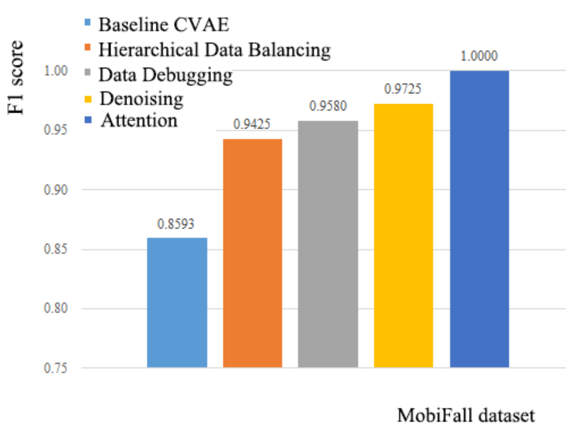


Fig. 8. The ablation study of the proposed fall detection method when using the MobiFall dataset

For this section, we conducted an ablation study with the three datasets to evaluate the effect of the proposed fall detection method. First, we measured the performance of the baseline CVAE model. Then, we investigated improvement in the accuracy of the CVAE model combined with the proposed data debugging and hierarchical data balancing techniques. Fig. 7 shows a side-by-side comparison of F1 scores for the proposed fall detection method when using the MobiAct dataset. Initially, the method employing just the baseline DCVAE model achieved an F1 score of 0.8673. Upon incorporation of hierarchical balancing, there was a noticeable improvement, raising the F1 score to 0.9137 — an increase of 0.046. Further advancements were observed when data debugging techniques were applied, enhancing the F1 score

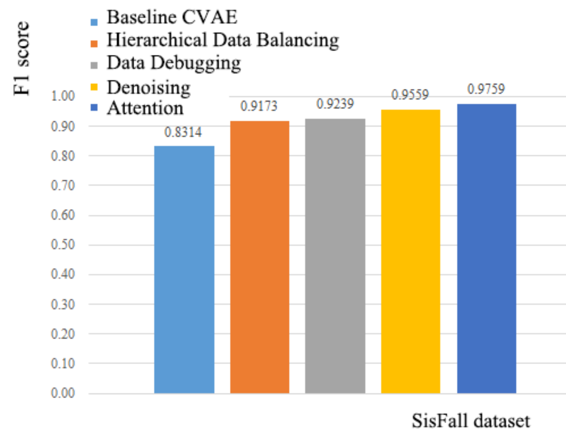


Fig. 9. The ablation study of the proposed fall detection method when using the SisFall dataset

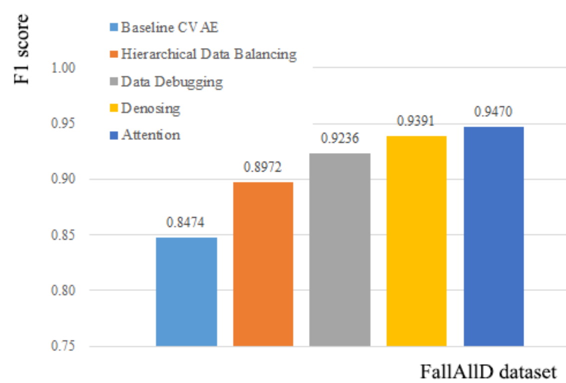


Fig. 10. The ablation study of the proposed fall detection method when using the FallAllD dataset

by 0.023 to reach 0.9373. Implementing denoising training methods also led to better results, with the F1 score climbing to 0.9589 and an incremental gain of 0.021. Adding an attention layer to the CVAE model achieved substantial enhancement. It propelled the F1 score to a remarkable 0.9953, marking a significant leap of 0.036 compared to the previous method. Fig. 8 shows the F1 scores of the proposed fall detection method on the MobiFall dataset. The baseline CVAE model starts with an F1 score of 0.8593. Using the hierarchical data balancing technique, the score of 0.9425 increases by approximately 0.0832. By applying the data debugging techniques, we can see an improvement in the F1 score, which is higher at 0.0155, achieving an F1 score of 0.9580. The denoising training approach shows a slight increase in performance with an F1 score of 0.9725. Finally, a significant enhancement was achieved by adding an attention layer to the CVAE model. It propelled the F1 score to a remarkable 1.0. Fig. 9 shows the F1 scores of the proposed fall detection method on the SisFall dataset. The baseline CVAE model started with an F1 score of 0.8314. We obtained an F1 score of 0.9173 by using hierarchical data balancing, increasing it by 0.0859.

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.2024.3388478

By applying data debugging, we see an F1 score of 0.0066 improvement to reach 0.9239. The denoising training approach showed a slight increase in performance, obtaining an F1 score of 0.9559. Finally, a significant enhancement was achieved by adding an attention layer to the CVAE model. That propelled the F1 score to a remarkable 0.9759. Fig. 10 shows the F1 scores of the proposed fall detection method on the FallAllD dataset. The baseline CVAE model had an initial F1 score of 0.8474. We were able to increase it to 0.8972 using hierarchical data balancing. By applying data debugging, we observed an improvement of 0.0264 to reach an F1 score of 0.9239. The denoising training approach slightly improved, obtaining an F1 score of 0.9319. Finally, adding an attention layer to the CVAE model significantly enhanced the F1 score to an impressive 0.9470. Based on these results, the proposed data debugging and hierarchical data balancing techniques have significantly improved the accuracy of the fall detection system.

V. Conclusion

Despite significant advancements in fall detection systems, there are inherent problems with data shortages and limited resource allocations in wearable devices. To address these issues, we proposed a novel fall detection method using a DCVAE, applying data debugging and hierarchical data balancing techniques, diverging from traditional methods that depend heavily on complex DL models with many parameters. As a result, the proposed unsupervised learning fall detection method achieved F1 scores of 1.0.

The proposed method solves the lack of data problem by only training with ADL data. Moreover, the DCVAE model used in the proposed fall detection method performed better and has $25.6\times$ fewer parameters than the model in [30]. That demonstrated that our method performs better and is more memory-efficient than the approach proposed in [30], making it highly suitable for wearable devices.

A considerable limitation of our work is that the proposed fall detection method primarily identifies falls after they occur, limiting the opportunity for immediate preventive actions to mitigate injury risks. We need pre-impact [4] or near-fall detection monitoring to overcome this limitation, which allows for timely interventions to prevent the fall and its associated injuries. Hence, in future work, we plan to consider advanced unsupervised learning, such as the diffusion model, to extend our work to complex situations, such as pre-impact falls and near-falls, which may prove much more effective in preventing falls.

References

[1] Minming Gu, Zhixiang Chen, Kaiyu Chen, and Haipeng Pan, ”IR-ST: A Lightweight Transformer Network for Human Fall Detection Based on FMCW Radar,” *IEEE Sensors Journal*, Vol. 23(20), pp. 25128 - 25135, 2023

[2] Rebecca J. Tarbert and Wamis Singhatat, ”Real World Evidence of Wearable Smartbelt for Mitigation of Fall Impact in Older Adult Care,” *IEEE Journal of Translational Engineering in Health and Medicine*, Vol.11, pp. 247 - 251, 2023

[3] Ke Zhang, Wenbo Liu, Jingsha Zhangm, Zengyong Li, and Jizhong Liu, ” A Fall Risk Assessment Model for Community-Dwelling Elderly Individuals Based on Gait Parameters,” *IEEE Access*, Vol. 11, pp. 120857 - 120867, 1012

[4] Rahul Jain and Vijay Bhaskar Semwal, ”A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors,” *IEEE Sensors Journal*, Vol. 22(23), pp. 22943-22951, 2022

[5] Vijay Bhaskar Semwal, Abhishek Kumar, Pankaj Nargesh, and Vaibhav Soni Semwal, ”Tracking of Fall Detection Using IMU Sensor: An IoHT Application.”, Machine Learning, Image Processing, Network Security and Data Sciences, Lecture Notes in Electrical Engineering, Vol. 946, 2023

[6] Xueyi Wang, Joshua Ellul, and George Azzopardi, ”Elderly Fall Detection Systems: A Literature Survey,” *Frontiers in Robotics and AI*, Vol. 7, 2020

[7] Vasileios-Rafail Xefteris, Athina Tsanousa, Georgios Meditskos, Stefanos Vrochidis, and Ioannis Kompatsiaris, ”Performance, Challenges, and Limitations in Multimodal Fall Detection Systems: A Review,” *IEEE Sensors Journal*, Vol. 21(17), pp. 18398 - 18409, September 2021

[8] Vijay Bhaskar Semwal, Praveen Lalwani, Manas Kumar Mishra, Vishwanath Bijalwan, and Jasroop Singh Chadha, ”An optimized feature selection using bio-geography optimization technique for human walking activities recognition,” *Computing* , Vol. 103, pp. 2893-2914, 2021

[9] M. Saleh, M. Abbas and R. L. B. Jeannès, ”FallAllD: An Open Dataset of Human Falls and Activities of Daily Living for Classical and Deep Learning Applications,” *IEEE Sensors Journal*, Vol. 21(2), pp. 1849 - 1858, 2020.

[10] Xiaoqun Yu, Jaehyuk Jang, and Shuping Xiong, ”A Large-Scale Open Motion Dataset (KFall) and Benchmark Algorithms for Detecting Pre-impact Fall of the Elderly Using Wearable Inertial Sensors, ,” *Frontiers in Aging Neuroscience*, Vol. 13, 2021

[11] Vijay Bhaskar Semwal, Anjali Gupta, and Praveen Lalwani, ”An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition,” *The Journal of Supercomputing*, Vol. 77, pp. 12256-12279, 2021

[12] Manoj Kumar Sain, Joyeeta Singha, Sandeep Saini, and Vijay Bhaskar Semwal, ”Human Action Recognition Using ConvBiLSTM-GRU In Indoor Environment,” In Proc. of the 2023 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT). 2023

[13] Sravan Kumar Challa, Akhilesh Kumar, Vijay Bhaskar Semwal, and Nidhi Dua, ”An optimized deep learning model for human activity recognition using inertial measurement units,” *Expert Systems*, Vol. 40(10), e13457, 2023

[14] Vijay Bhaskar Semwal, Rahul Jain, Pushkar Maheshwari, and Saksham Khatwani, ”Gait reference trajectory generation at different walking speeds using LSTM and CNN”,” *Multimedia Tools and Applications*, Vol. 82, pp. 33401–33419, 2023

[15] Sravan Kumar Challa, Akhilesh Kumar, Vijay Bhaskar Semwal, and Nidhi Duam, ”An optimized-LSTM and RGB-D sensor-based human gait trajectory generator for bipedal robot walking,” *IEEE Sensors Journal*, Vol. 22(24), pp. 24352-24363, 2022

[16] Semwal Vijay Bhaskar, Pavan Chakraborty, and Gora Chand Nandi, ”Less computationally intensive fuzzy logic (type-1)-based controller for humanoid push recovery,” *Robotics and Autonomous Systems*, Vol. 63 pp.122-135, 2015.

[17] Vijay Bhaskar Semwal and Gora Chand Nandi. ”Toward Developing a Computational Model for Bipedal Push Recovery-A Brief.” IEEE Sensors Journal, Vol. 15(4) pp. 2021-2022, 2015

[18] Vijay Bhaskar Semwal and G.C. Nandi, ”Study of humanoid push recovery based on experiments,” In Proc. of the IEEE 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE). 2013

[19] Vijay Bhaskar Semwal,” *Data driven computational model for bipedal walking and push recovery,”*, Unpublised doctoral dissertation, Indian Institute of Information Technology, arXiv preprint arXiv:1710.06548, 2017

[20] Vijay Bhaskar Semwal, Pratibha Tokas, Maske Nilesh, and Nilay Khare, ”Dynamic balancing of bipedal robot on inclined terrain using polynomial trajectories,” In Proc. of the 2023 7th International Conference on Graphics and Signal Processing, 2023

[21] Vijay Bhaskar Semwal, Youngshik Kim, Vishwanath Bijalwan, Astha Verma, Ghanapriya Singh, Neha Gaud, Hangyeol Baek, and Abdul Manan Khan, ”Development of the LSTM Model and Universal Polynomial Equation for all the Sub-phases of Human Gait,” *IEEE Sensors Journal*, Vol. 23(14), pp.15892 - 15900, 2023

[22] Vijay Bhaskar Semwal, Kaushik Mondal, and Gora Chand Nandi, ”Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach,” *Neural Computing and Applications*, Vol. 28, pp. 565–574, 2017

[23] Lian Wu, Chao Huang, Shuping Zhao, Jinkai Li, Jianchuan Zhao, Zhongwei Cui, Zhen Yu, Yong Xu, and Min Zhang, “Robust fall detection in video surveillance based on weakly supervised learning Neural Networks”, Vol. 163, pp. 286-297, June 2023

[24] Van-Ha Hoang, Jong Weon Lee, Md. Jalil Piran, and Chun-Su Park, ”Advances in Skeleton-Based Fall Detection in RGB Videos: From Handcrafted to Deep Learning Approaches,” *IEEE Access*, Vol.11, pp. 92322 - 92352, 2023

[25] Jing Zhang, Jia Li, and Weibing Wang, ”A Class-Imbalanced Deep Learning Fall Detection Algorithm Using Wearable Sensors,” *Sensor*, Vol. 21(19), 2021,

[26] Myung-Kyu Yi, Wai-Kong Lee, and Seong Oun Hwang, “A Human Activity Recognition Method Based on Lightweight Feature Extraction Combined With Pruned and Quantized CNN for Wearable Device,” *IEEE Transactions on Consumer Electronics*, Vol. 69(3), pp. 657 - 670, August 2023.

[27] https://www.st.com/en/evaluation-tools/nucleo-h755zi-q.html

[28] Mostafa Rahimi Azghadi, Corey Lammie, Jason K. Eshraghian, Melika Payvand, Elisa Donati, Bernabé Linares-Barranco, and Giacomo Indiveri, “Hardware Implementation of Deep Network Accelerators Towards Healthcare and Biomedical Applications,” *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 14(6), pp.1138 - 1159, December 2020.

[29] Xu Yang, Chen Zhuang, Wenquan Feng, Zhe Yang, and Qiang Wang, ”FPGA Implementation of a Deep Learning Acceleration Core Architecture for Image Target Detection,”*Applied Sciences*, Vol. 13(7), 2023

[30] Young-Hoon Nho, Semin Ryu, and Dong-Soo Kwon, ”UI-GAN: Generative Adversarial Network-Based Anomaly Detection Using User Initial Information for Wearable Devices,” *IEEE Sensors Journal*, Vol. 21(8), pp. 9949 - 9958, 2021.

[31] Sahaj Anilbhai Patel and Abidin Yildirim, “Non-stationary neural signal to image conversion framework for image-based deep learning algorithms,” *Frontiers in Neuroinformatics*, Vol. 17, 2023

[32] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection,” In Proceedings of the International Joint Conf. Neural Netw. (IJCNN), pp. 1–8, 2019.

[33] Che, Tong, Li, Yanran, Jacob, Athul Paul, Bengio, Yoshua, and Li, Wenjie, “Mode regularized generative adversarial networks,”In International Conference on Learning Representations(ICLR), 2017.

[34] Eduardo Casilari, José-Antonio Santoyo-Ramónm and José-Manuel Cano-García, ”Analysis of Public Datasets for Wearable Fall Detection Systems,” *Sensors*, Vol. 17(7), 2017

[35] https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html

[36] James Ren Hou Lee, Linda Wang, and Alexander Wong, ”EmotionNet Nano: An Efficient Deep Convolutional Neural Network Design for Real-time Facial Expression Recognition, Computer Vision and Pattern Recognition, 2022.

[37] Bart Cox, Robert Birke, and Lydia Y. Chen, ”Memory-aware and context-aware multi-DNN inference on the edge,” *Pervasive and Mobile Computing*, Vol. 83, 2022

[38] Sucerquia A., López J.D., Vargas-Bonilla J.F., ”SisFall: A fall and movement dataset”, *Sensors*, Vol. 17, 2017

[39] Kukjin Choi, Jihun Yi, And Changhwa Park, ”Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis”,*IEEE ACCESS*, Vol. 9, pp. 120043 - 120065, 2021.

[40] Diederik P. Kingma and Max Welling, ”An Introduction to Variational Autoencoders”, Foundations and Trends in Machine Learning: Vol. 12: No. 4, pp 307-392. 2019

[41] Bruce Ratner, ”The correlation coefficient: Its values range between +1 and -1, or do they?,*Journal of Targeting, Measurement and Analysis for Marketing*, Vol. 17. pp.139–142, 2009

[42] D. Bahdanau, K. Cho and Y. Bengio, ”Neural machine translation by jointly learning to align and translate” in arXiv:1409.0473, 2014

[43] Justus Zipfel, Felix Verworner, Marco Fischer, Uwe Wieland, Mathias Kraus, and Patrick Zschech, ”Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning model,” *Computers & Industrial Engineering*, Vol. 177, March 2023.

[44] G. Vavoulas, M. Pediaditis, E. G. Spanakis, and M. Tsiknakis, ”The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones,” in Proc. 13th IEEE Int. Conf. Bioinf. BioEng., pp. 1–4, Nov. 2013.

[45] A. Jahanjoo, M. Naderan, and M. J. Rashti, “Detection and multiclass classification of falling in elderly people by deep belief network algorithms,” *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1–21, Jan. 2020.

[46] C. Chatzaki, M. Pediaditis, G. Vavoulas and M. Tsiknakis, “Human daily activity and fall recognition using a smartphone’s acceleration sensor” in Information and Communication Technologies for Ageing Well and e-Health, Springer, vol. 736, 2017.

[47] Supannada Chotipant and Pornsuree Jamsri, ”Differentiative Feature-based Fall Detection System,” In Proceedings of the 21st International Conference on Control, Automation and Systems (ICCAS), October 2021.

[48] Xiaodan Wu, Yumeng Zheng, Chao-Hsien Chu, Lingyu Cheng, and Jungyoon Kim,”Applying deep learning technology for automatic fall detection using mobile sensors,” *Biomedical Signal Processing and Control*, Vol. 72, Part B, February 2022.

[49] Hong-Lam Le, Duc-Nhan Nguyen, Thi-Hau Nguyen,and Ha-Nam Nguyen, “A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System,” *Electronics*, Vol. 11(7), 1030, 2022.

[50] Yanli Li, Zhengwei Zuo, and Julong Pan, ”Sensor-based fall detection using a combination model of a temporal convolutional network and a gated recurrent unit,” *Future Generation Computer Systems*, Vol. 139, pp. 53-63, 2023

[51] Eduardo Casilari, Moisés Álvarez-Marcom, and Francisco García-Lagos, ”Study of the Use of Gyroscope Measurements in Wearable Fall Detection Systems,” *Symmetry*, Vol. 12(4), 2020

[52] Xu, J., He, Z., and Zhang, Y., ”CNN-LSTM Combined Network for IoT Enabled Fall Detection Applications,” In Proceedings of the Journal of Physics: Conference Series, Xi’an, China, 25–27 April IOP , Publishing, Vol., p. 012044, 2019

[53] Haoran Zhang and Linhai Xu, ”Multi-STMT: Multi-Level Network for Human Activity Recognition Based on Wearable Sensors,” *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, Vol. 73, 2508612, 2024

[54] Chien-Pin Liu, Ju-Hsuan Li, En-Ping Chu, Chia-Yeh Hsieh, Kai-Chun Liu, Chia-Tai Chan, Yu Tsao, ”Deep Learning-based Fall Detection Algorithm Using Ensemble Model of Coarse-fine CNN and GRU Networks,” In Proc. of the 2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA), 2023

Myung-Kyu Yi (Member, IEEE) received a Ph.D. in Computer Science and Engineering from Korea University in 2005. He is currently a Research Professor with Gachon University, and is a member of the personal health information standardization task force of the TTA U-Health project group. His research interests include healthcare, security, machine learning, deep learning, and human activity recognition.

KyungHyun Han received a B.Sc degree in computer engineering, an M.S. degree in electronics and computer engineering, and a Ph.D. degree in electronics and computer engineering from Hongik University, South Korea, in 2015, 2017, and 2023, respectively. He is currently a Researcher with the Information Security and Machine Learning Laboratory at Gachon University in South Korea. His research interests include cyber security, GPU computing, machine learning, and blockchain.

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.2024.3388478

Seong Oun Hwang (Senior Member, IEEE) received the B.S. degree in mathematics from Seoul National University, in 1993, the M.S. degree in information and communications engineering from the Pohang University of Science and Technology, in 1998, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology, South Korea. He worked as a Software Engineer with LG-CNS Systems, Inc., from 1994 to 1996. He also worked as a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), from 1998 to 2007. He worked as a Professor with the Department of Software and Communications Engineering, Hongik University, from 2008 to 2019. He is currently a Professor with the Department of Computer Engineering, Gachon University. His research interests include cryptography, cybersecurity, and artificial intelligence. He is an Editor of ETRI Journal.