information

MDPI

Article

Identification of Optimal Data Augmentation Techniques for Multimodal Time-Series Sensory Data: A Framework

Nazish Ashfaq [1], **Muhammad Hassan Khan** [1,*] and **Muhammad Adeel Nisar** [2]

[1] Department of Computer Science, University of the Punjab, Lahore 54590, Pakistan; phdcsf21m510@pucit.edu.pk
[2] Department of Information Technology, University of the Punjab, Lahore 54000, Pakistan; adeel.nisar@pucit.edu.pk
* Correspondence: hassankhan@pucit.edu.pk

Abstract: Recently, the research community has shown significant interest in the continuous temporal data obtained from motion sensors in wearable devices. These data are useful for classifying and analysing different human activities in many application areas such as healthcare, sports and surveillance. The literature has presented a multitude of deep learning models that aim to derive a suitable feature representation from temporal sensory input. However, the presence of a substantial quantity of annotated training data is crucial to adequately train the deep networks. Nevertheless, the data originating from the wearable devices are vast but ineffective due to a lack of labels which hinders our ability to train the models with optimal efficiency. This phenomenon leads to the model experiencing overfitting. The contribution of the proposed research is twofold: firstly, it involves a systematic evaluation of fifteen different augmentation strategies to solve the inadequacy problem of labeled data which plays a critical role in the classification tasks. Secondly, it introduces an automatic feature-learning technique proposing a Multi-Branch Hybrid Conv-LSTM network to classify human activities of daily living using multimodal data of different wearable smart devices. The objective of this study is to introduce an ensemble deep model that effectively captures intricate patterns and interdependencies within temporal data. The term "ensemble model" pertains to fusion of distinct deep models, with the objective of leveraging their own strengths and capabilities to develop a solution that is more robust and efficient. A comprehensive assessment of ensemble models is conducted using data-augmentation techniques on two prominent benchmark datasets: CogAge and UniMiB-SHAR. The proposed network employs a range of data-augmentation methods to improve the accuracy of atomic and composite activities. This results in a 5% increase in accuracy for composite activities and a 30% increase for atomic activities.

Keywords: data augmentation; ensemble deep network; human activity recognition; multimodal time series data

check for updates

Citation: Ashfaq, N.; Khan, M.H.; Nisar, M.A. Identification of Optimal Data Augmentation Techniques for Multimodal Time-Series Sensory Data: A Framework. *Information* **2024**, *15*, 343. https://doi.org/10.3390/info15060343

Academic Editors: Gholamreza Anbarjafari, Jörg Schäfer and Matthias Wagner

Received: 17 April 2024
Revised: 2 June 2024
Accepted: 3 June 2024
Published: 11 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

People in today's world are very accustomed to using wearable gadgets such as smart phones, watches and eye-wear. Typically, these devices are equipped with numerous sensing modalities such as inertial measurement units (IMUs), position sensors, ambient light sensors, proximity sensors, etc., that produce massive amounts of data every day [1]. These data can be used in various activity recognition applications, especially in healthcare, such as geriatric monitoring, to recognise and assess the actions of the elderly in order to forecast future health issues [2]. Other than monitoring daily physical activity levels, they may also be used to suggest healthier exercise regimens. Moreover, they can be utilised to perform activity analysis on patients undergoing post-operative rehabilitation to provide doctors with a more thorough comprehension of their current state, therefore expediting patient evaluation and care [3].

This extensive usage of sensing modalities allows consistent accumulation of diverse forms of data. These sensing modalities have the capability to quantify many parameters such as temperature, motion, sound and even heart rate [4]. For example, a smartphone has the capability to monitor the number of steps you take within a day through accelerometer sensors or document the temperature of your environment through environment sensors. Smartwatches have the capability to track your heart rate via LEDs and optical sensors while you are engaged in physical activity, whereas smart glasses can offer up-to-the-minute data about your surroundings. When all of these data are processed using machine learning algorithms, it creates a comprehensive picture of our environment and daily life, providing previously unobtainable insights [5].

This large amount and intricate nature of sensory information present considerable obstacles for thorough analysis and understanding. Conventional analytical techniques frequently face challenges in deriving useful insights from these data flows, resulting in the under-utilisation of these data. Deep learning models have shown great promise in a number of fields, including human activity recognition [6], healthcare [7] and elderly care [8]. They have been more popular recently for handling time series-related tasks like classifying time series data [9], forecasting future values [10] and identifying abnormalities in time series [11]. Having a large amount of labeled training data is essential for deep learning to be successful [12]. Nonetheless, the obtained data from these smart gadgets are huge but unusable because they are unlabeled, and we lack enough labeled instances to train the classification models efficiently. This issue causes the model to become overfit. To solve this inadequacy of labeled data problem, we need a large amount of labeled data which requires manual labeling of sensor data through a costly, time-consuming and tedious process. Synthetic data generation using data augmentation has been one approach to obtain additional information over the last two decades [13,14]. Notably, data augmentation is a general-purpose data side solution that is independent of the input space of a deep learning model yet maintains correct labels. Data augmentation aims to decrease overfitting and broaden the model's decision boundary so as to improve the model's capacity for generalisation [15]. In real-world data, the need for generalisation is particularly crucial and can aid networks in overcoming small datasets [16] or datasets with unequal class distributions [17,18].

Data augmentation is a well-known phenomena in the domain of digital images. Most of the early cutting-edge Convolutional Neural Network (CNN) [19] architectures used data augmentation, such as cropping [20], scaling [21], mirroring [22] and colour augmentation on images [23–25]. Although data augmentation is frequently used in neural network-based image identification, it is not a recognised best practice for time series recognition [26]. Compared to data augmentation for images, stochastic transformations of the training data for the time series data have not been explored thoroughly. For instance, some techniques that have been employed on time series data adequately include introducing arbitrary noise, cutting or resising, adjusting the scale, applying random distortions in the temporal dimension [27–30] and modifying the frequency characteristics [31]; however, numerous other techniques such as augmix, hide and seek, mixup and cutmix, etc., have been explored on digital images but not on time series data [32,33]. The above-mentioned references provide a very interesting and explorable research gap. One challenge associated with data augmentation based on random transformations is the presence of a wide variety of time series modalities, each possessing unique characteristics.

We have performed a systematic evaluation of numerous augmentation techniques on time series multimodal sensory data related to activities of daily living (ADLs). For this purpose two datasets of ADLs, namely CogAge [34] and UniMiB-SHAR [35], are used for the identification of useful methods for these activities. ADLs can be divided into two groups based on activities that people do on a daily basis, namely short-term (i.e., atomic) and long-term (i.e., composite) [36]. Long-term activities like cooking, brushing teeth, cleaning hands, etc., can be categorised as composite activities, whereas short bursts of movement like lifting an arm or a leg are referred to as atomic activities [37]. Atomic actions are further divided into two categories: behaviour (such as sitting down, standing up, and lying down) and position (such as sitting, standing, and lying) [34]. The limitation of these approaches is that they are less effective when applied to composite activity data and they lose their effectiveness in the presence of noisy data. This paper focuses not only on classification of activities of daily living but also presents various data-augmentation approaches for time series multimodal sensory data. We present an ensemble model also termed hybrid model involving the combination of distinct deep models with the goal of leveraging their individual strengths and capabilities to create a stronger and more efficient solution. The proposed Multi-Branch Hybrid Conv-LSTM (MHyCoL) model consists of two convolution blocks, each following a pooling layer, a flatten layer, a dropout layer and a dense layer in each of its adaptive branches and two LSTM blocks followed by two dense layers. The branched CNN model operates simultaneously with changeable input to work with time series multimodal sensory data. Each CNN branch in the model corresponds to a unique sensor modality having different frequencies. CNNs are used to capture local patterns and spatial information in our temporal data, while Long Short-Term Memory (LSTM) networks are used to capture long-range dependencies and temporal dynamics, making them suitable for sequential data classification. We have applied fifteen different data-augmentation techniques over the atomic and composite activities and enhanced accuracy by 5% in composite activities and 30% in atomic activities. The major contributions of the proposed research are explained in the following:

- A systematic evaluation of different augmentation techniques is presented to solve the inadequacy problem of labeled time series data.
- An automatic feature-learning technique is proposed to recognise multimodal data of different wearable smart devices.
- A detailed overview of existing techniques and their categorisation is presented.
- An extensive experimental evaluation is proposed on two benchmark datasets.

The rest of the paper is organised as follows: An extensive review of previous work is provided in Section 2, and a detailed explanation of the proposed model and augmentation methods is covered in Section 3. Section 4 covers all details of the experiments carried out for the purpose of research. This study is finally concluded in Section 5, which offers some final thoughts and suggests some possible directions for further research.

2. Literature Review

Human activity detection has stimulated the interest of researchers in recent years due to its applications in physical health evaluation in rehabilitation facilities, suspicious activity recognition in the context of security and gesture recognition in video games. The actions of a person's daily life are lengthy and typically performed in a hierarchical order. Identifying human long-term activities is thus a hierarchical task. Several studies have been undertaken in recent years to recognise the longer chronological human activity. The literature review is divided into two sections. In the first section, we give a brief review of activity-recognition techniques and the second section emphasises different augmentation techniques.

2.1. Human Activity Recognition

Time series raw sensory data in its nature is a representation of information collected over time, such as sequential events, recording trends and patterns. It is necessary to encode features in order to transform raw data into a processable format for machine learning algorithms, hence enabling them to recognise temporal patterns and relationships within the data [38]. For feature engineering of the raw data, the most commonly used step is windowing of time series sequences. Windowing is a process that adds temporal context into feature engineering. By using data within a window, we capture dependencies and patterns over time. The existing techniques for human activity recognition (HAR) are classified into three categories: (1) handcrafted feature-based techniques (HFTs), (2) codebook-based feature encoding techniques (CBTs) and (3) automatic feature-learning techniques (AFTs). Each category is discussed in the following subsections.

2.1.1. Handcrafted Feature-Based Techniques

Handcrafted features refer to manual or engineered properties obtained from raw data that aim to capture specific patterns related to the task at hand. In HFTs, statistical measures such as mean, variance, standard deviation, percentiles, Fourier transformations or wavelets, etc., are applied on preprocessed sensory data to form a feature vector to be fed to classification algorithms written for human activity recognition. The whole process of handcrafted feature computation and classification is depicted in Figure 1.

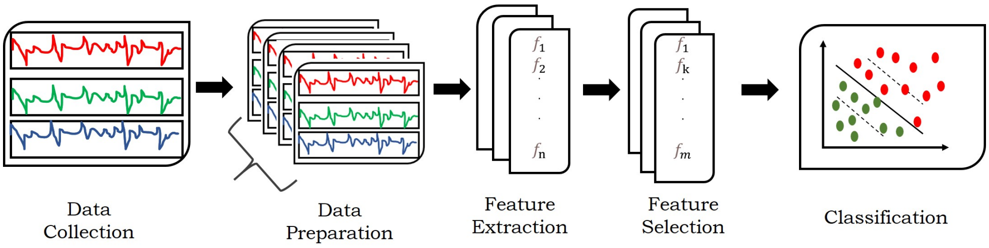

Figure 1. An illustration exhibiting the process of detecting human actions utilising raw sensory data using handcrafted feature-based encoding approaches.

Amjad et al. [39] proposed a two-level hierarchical method for HAR using wearable sensors. Firstly, they detected atomic activities using 17 handcrafted features including mean, variance, skewness, first-order norm, etc. Secondly, these atomic actions were used to recognise composite activities. Similarly, Sargano et al. [40] employed various handcrafted feature-extraction techniques, including space-time, local binary patterns, appearance-based and fuzzy logic-based algorithms, on sensory input. Subsequently, these traits were combined and employed to train a classifier with the objective of achieving recognition. Hsu et al. [41] examined the patients' movement patterns by computing the bandwidth frequency, skewness and kurtosis features using time series sensory data. The effectiveness of HFT greatly depends on the researchers' expertise in the desired domain and their capacity to record significant information from the unprocessed data [42,43]. HFTs are more effective for short-term activities as they create features out of these sequences but they are unable to capture temporal sequences of long-term activities [34].

2.1.2. Codebook-Based Feature-Encoding Techniques

CBTs employ the clustering of similar patterns within the data in order to generate a codebook [44]. Every cluster corresponds to a unique pattern or characteristic. The process of quantising the data into representative clusters allows a reduction in the dimensionality of the time series, while still retaining pertinent information. This method is particularly advantageous for rapidly extracting significant characteristics from sensory data that is both high-dimensional and noisy. The whole process of codebook-based feature computation and classification is depicted in Figure 2.

Lagodzinski et al. [45] employed a codebook-based feature-extraction approach on the IMU data from smart glasses to identify behaviours such as reading from a printed page, drinking water, viewing a video and so on. A multilevel approach was proposed by Nisar et al. [34] to identify activities of everyday living. Atomic activities are identified at the first level of their framework by applying the codebook [46] approach and at the second level of their framework, composite activities are identified by using the rank pooling strategy based on the recognition scores of atomic activities. Koping et al. [47] introduced a feature-learning approach based on codebook for the purpose of recognising human actions using sensory data. The researchers employed the k-means clustering approach to generate a codebook and subsequently created a histogram-based feature vector representation by predicting the codewords within the activity sequences. One major limitation of the codebook-based approach is that its computation with optimal size of clusters is a complex process and requires a lot of time [48,49].

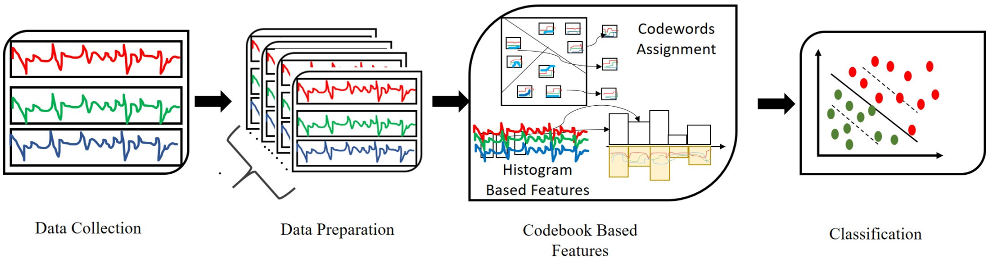

Figure 2. An illustration exhibiting the process of detecting human actions utilising raw sensory data using codebook feature-based encoding approaches.

2.1.3. Automatic Feature-Based Techniques

Numerous automatic feature-learning techniques have been explored to recognise human behaviours using sensory data by employing deep neural networks. The whole process of automatic feature computation and classification is depicted in Figure 3.

Zhang et al. [50] used commercially available wifi devices to compute the spectrum of ten human activities using a dense LSTM model. Some are atomic activities, such as walking and running, while others are composite, such as playing a guitar and playing basketball; they revealed the difference in recognition accuracy between atomic and composite activities. Bianchi et al. [51] presented a deep CNN. Anagnostis et al. [52] suggested a method to gather movement data from the human body using five IMU sensors; later, they employed an LSTM-based deep neural network to identify the actions of the subject in agricultural environments. Bu et al. [53] introduced a CNN with the aim to predict human behaviours. This was achieved by dynamically localising a limited number of activity-discriminative intervals, as opposed to using a fixed-length window. Cheng et al. [54] introduced a prototype-guided framework for activity recognition in order to effectively decouple the feature representation and classifier, hence providing support for Federated Learning in the context of data privacy. Huang et al. [55] introduced channel equalisation as a means to mitigate the interference caused by inhibited channels. This was achieved through the implementation of a whitening operation. This technique guarantees that all channels systematically contribute to the representation of features.

The ensemble deep network combines multiple individual models into one model. The primary objective of these networks is to enhance forecast accuracy, resilience and generalisability through the utilisation of the combined knowledge possessed by numerous models [56]. For example, LSTM-CNN models utilise the spatial pattern-capturing capabilities of CNNs in conjunction with the temporal sequence learning properties of LSTMs, leading to enhanced and precise recognition of human activities. Kolkar et al. [57] presented the hybrid CNN-GRU model for activity recognition. The initial stage of the model involves passing the input data through the CNN, which is then followed by dropout and pooling layers. In the second phase, the output of the CNN layers is inputted into GRU layers. Afterwards, the output of the GRU is categorised using a softmax layer to identify activities. Dua et al. [58] introduced a CNN-GRU ensemble model for the purpose of identifying human activities. Khatun et al. [59] introduced a combination of LSTM and CNN networks to identify human actions based on sensory input from smartphones. The authors determined that incorporating the self-attention mechanism enabled the architecture to concentrate on the most significant and pertinent elements, resulting in enhanced accuracy. The efficacy of ensemble models in capturing subtle temporal interactions and complex spatial features has been well-established [60]. Therefore, they have the potential to surpass the constraints of individual deep learning methods and provide a more comprehensive answer to the intricacy of HAR systems.

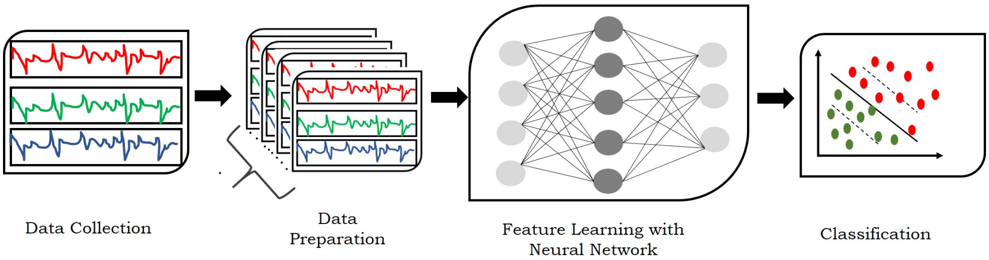

Figure 3. An illustration exhibiting the process of detecting human actions utilising raw sensory data using an automatic feature-learning approach.

A summary of different feature-learning techniques for human activity recognition is provided in Table 1.

Table 1. Summary of different feature-learning techniques for human activity recognition. Here sp, sw and sg denotes smartphone, smartwatch and smartglasses, respectively.

Ref.	Activity Type	Modality Type	Features
Amjad et al. [39]	atomic	sp, sw, sg	Handcrafted
Sargano et al. [40]	atomic	videos	Handcrafted
Hsu et al. [41]	atomic	IMU sensor	Handcrafted
Lagodzinski et al. [45]	composite	sg	Codebook
Nisar et al. [34]	atomic, composite	sp, sw, sg	Codebook
Koping et al. [47]	composite	sp, sw, sg	Codebook
Zhang et al. [50]	atomic, composite	wifi devices	LSTM
Bianchi et al. [51]	composite	IMUs	CNN
Anagnostis et al. [52]	atomic	IMUs	LSTM
Bu et al. [53]	atomic	IMUs	CNN
Huang et al. [55]	atomic	IMUs	CNN
Kolkar et al. [57]	composite	IMUs	CNN + GRU
Dua et al. [58]	atomic	IMUs	CNN + GRU
Khatun et al. [59]	atomic	IMUs	CNN + LSTM
Nisar et al. [61]	atomic, composite	sp, sw, sg	CNN + LSTM

2.2. Data Augmentation

Data augmentation is a beneficial method for enlarging the training dataset by implementing diverse alterations to the current data. Many earlier techniques for augmenting time series data, including cropping, inverting and noise addition, were derived from image data augmentation [27–29]. In general, time series transformations can be categorised into three distinct domains: magnitude, duration and frequency. Time series are transformed in the magnitude domain along the value or variation axes. Frequency domain transformations distort frequencies, whereas time domain transformations affect time increments. Additionally, hybrid methods which employ fusion of multiple domains also exist. It is important to acknowledge that the dataset can be aggregated using multiple transformation techniques, both sequentially [62] and concurrently [63,64]. The subsequent subsections will provide comprehensive descriptions of the random transformation-based data-augmentation methods and the pattern mixing method that are associated with each of these domains.

2.2.1. Magnitude Domain Transformations (MDTs)

Data augmentation in the MDT involves changing the values of the time series while keeping the time steps constant. These modifications only alter the values of each element, which is essential for preserving temporal integrity. Jittering is the most common data-transformation technique used for time series sensory data. For instance, Rashid et al. [64] enhanced the precision of LSTM for sensor data originating from construction equipment by combining jittering with additional data-augmentation techniques. Um et al. [62] applied jittering to wearable sensor data for Parkinson's disease monitoring using ResNet. Steven et al. [65] utilised Gaussian noise in conjunction with various amplification techniques to process atomic activity using sensor data. Rashid et al. [64] applied flipping to univariate time series. Although rotation data augmentation is effective in generating realistic patterns for image identification, it may not be appropriate for time series data as rotating a time series can alter the class label assigned to the original sample [66]. Rotation augmentation has been observed to either have no impact or a negative impact on time series categorisation when using neural networks [64,67,68]. In contrast, Um et al. [62] discovered that using rotation data augmentation resulted in enhanced accuracy, particularly when used in conjunction with other augmentation techniques. Tran and Choi [69] employed a technique that involved combining scaling with jittering and element-wise interpolation for the purpose of gait identification. Tsinganos et al. [70] utilised surface electromyography (sEMG) data and implemented various augmentation approaches, such as magnitude warping, to demonstrate the effectiveness of data augmentation in enhancing model correctness and generalisation capability.

These techniques are to capture variations in signal intensity, providing insights into amplitude-related patterns and helping to generalise models; their major weakness is the possible amplification of noise or distortion of characteristics of signals if transformation parameters are not carefully managed. This may lead to inaccurate model predictions or data loss [64].

2.2.2. Time Domain Transformations (TDTs)

In contrast to MDT, the TDT moves the elements along the timeline. This means that time series elements are moved to different time stamps from where they started. Jeong et al. [71] introduced a new technique for data augmentation called time warping and applied it to partially obscured data from the accelerometer signals. Cheng et al. [72] performed data augmentation (permuting, resampling) on HAR data to solve the inadequacy problem using contrasitive learning. Similar work was carried out by steven et al. [65], where they employed an ensemble of augmentation techniques (permuting, time warping etc.) and showed improvement in the results. Rashid et al. [64] applied time warping to univariate time series data. Uchitomi et al. [73] employed time warping, cropping and permutation on Parkinson's disease data.

These techniques are to preserve temporal relationships allowing the model to capture important sequential patterns and dependencies for time series analysis; however, they may not capture nonlinear relationships or changes in signal characteristics limiting the model's ability to generalise complex temporal patterns [62,72].

2.2.3. Mixing Patterns (MPs)

MPs are the process of combining two or more patterns to create new ones. For random transformations, it is assumed that the transformation results are representative of the dataset. Not every transformation, however, is appropriate to every dataset. MPs presents a notable advantage as they avoid dependence on identical assumptions, and they embrace the notion that diverse patterns have the potential to be seamlessly integrated, resulting in advantageous outcomes [74]. Averaging two patterns can be used to generate new patterns. This technique was widely utilised for picture data augmentation (i.e. Mixup), in which they combined the channels of two images from the same class to create a new image [32]. Most reference patterns are randomly selected from the same class or utilising nearest neighbors. Numerous techniques from the category of MPs are employed such as cutmix, augmix, mixup, etc. Cutmix and similar techniques substitute arbitrarily shaped segments from one image for the other [75,76]. Gau et al. [74] employed these methods on time series data.

These techniques are capable of enhancing model robustness for varied data distributions by introducing variability through a combination of multiple augmentation strategies. However, they may result in additional computational burden and intricacy when adjusting settings for each augmentation technique, potentially resulting in longer training duration and higher resource demands. Furthermore, the inadequate integration of patterns may have a detrimental impact on the performance of the model if not executed with caution [74].

2.2.4. Deep Learning-Based Generative Models

Deep learning-based models, i.e., GAN [77,78], were also introduced for image data augmentation and they have gained significant popularity in recent times. Lou et al. [79] built a GAN using a fully connected network. They utilised an autoencoder network in conjunction with a Wasserstein GAN (WGAN) to enhance time series regression data. Chen et al. [80] introduced EmotionalGAN, a model that utilises 1D CNNs to classify emotions from extended ECG patterns. Significant improvements were discovered when data augmentation was applied to Support Vector Machines (SVM) and Random Forests. Fons et al. [81] proposed an automated data-augmentation technique, which focuses on time series data. They developed two automated weighting schemes that determine the contribution of augmented samples to the loss function. Additionally, one of the schemes selects a subset of transformations based on the predicted training loss ranking. Both adaptive policies show significant improvement in classifying various time series datasets. GAN has more time complexity, whereas random transformations are simple and less time-consuming approaches. Therefore, we analysed the strength of these approaches in our work.

Table 2 shows the summary of various types of data-augmentation techniques with reference to accuracy.

Table 2. Summary of various types of data-augmentation techniques with reference to accuracy. 'w' in the accuracy column represents augmentation (employed where difference of with and without was not provided) and 'T', 'M' and 'P' in the category column denote time domain, magnitude domain and mixing patterns respectively.

Reference	Augmentation Technique	Category	Dataset	Activity Type	Accuracy Increase
zhang et al. [50]	Time stratching, Spectrum shifting, Scaling, Frequency filtering, Jittring	T, M	CSI data	atomic	11%
Rashid et al. [64]	Jittering, Scaling, Rotation and Time warping	T, M	Equipment Activity Recognition	atomic	w 97.9%
Steven et al. [65]	Jittering, Permuting, Scaling, Rotating, Time-warping, Magnitude warping	T, M	UCI HAR	atomic	2%
Uchitomi et al. [73]	Rotation, Jittering, Scaling, Magnitude warping, Permutation, Time warping and Cropping	T, M	Parkinson's disease data	atomic	w 86.4%
huang et al. [82]	Linear interpolation.	T	WISDM	atomic	w 95.7%
oh et al. [83]	Linear interpolation	T	85 UCR Archive datasets		1%
cheng et al. [72]	Rotation, Jittering, Scaling, Permutation, Flipping and Resampling	T, M	UCR Archive datasets	atomic	5–10%
shi et al. [84]	Feature window	M	WISDM and MHEALTH	atomic	5%
guo et al. [74]	Manifold Mixup, Cutmix, Mixup, Cutout, Rotation, Jittering, Scaling, Permutation	T, M and P	5 UCR Archive datasets	atomic	2%

3. Proposed Method

This section provides a brief overview of the data-augmentation approaches used in comparative analysis to deal with the inadequacy problem of labeled data. The paper investigates three distinct data-augmentation categories. Later, it presents an ensemble model, namely MHyCoL which fuses the characteristics of both CNN and LSTM networks. The proposed model is evaluated using two benchmark datasets: CogAge and UniMib-SHAR. Figure 4 shows the flow of activity classification by our proposed work.

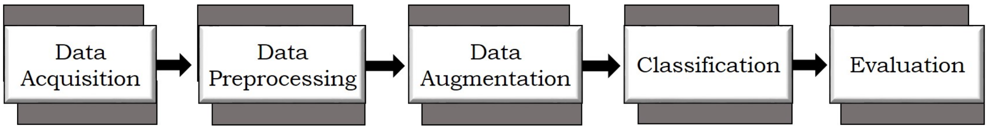

Figure 4. Flow of activity classification performed with our proposed approach.

3.1. Augmentation Techniques

This section provides the details of data-augmentation techniques employed on time series multimodal sensory data. These techniques are divided into three broad categories, namely magnitude domain transformations, time domain transformations and mixing patterns.

3.1.1. Augmentation Based on MDTs

The set of techniques in this domain involves the application of transformations to the values of time series. A crucial attribute of magnitude transformations is that they maintain constant time steps and modify only the values of each element.

- **Jittering:** This is a process of introducing noise to time series. It is an example of a transformation-based data-augmentation method that is both straightforward and efficient. Equation (1) represents mathematical notation of jittering.

$$\hat{X}_j = x_1 + \epsilon_1, \ldots, x_t + \epsilon_t, \ldots, x_T + \epsilon_T \tag{1}$$

where ϵ belongs to Gaussian noise (N) injected to each time step t and $\epsilon \in N(0, \sigma^2)$. The standard deviation σ^2 is set to 0.1. Figure 5 demonstrates the actual and transformed data after applying jittering on CogAge atomic (bending) activity.

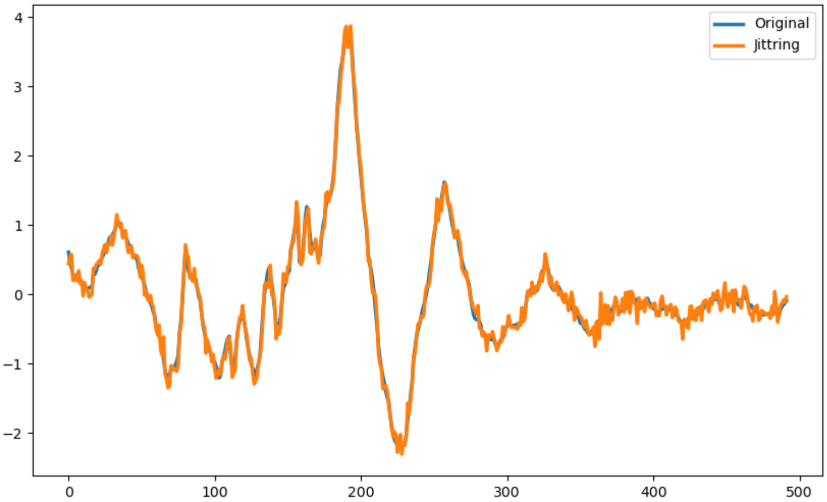

Figure 5. A visual representation of actual and transformed data after applying jittering on a CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

- **Scaling:** A random scalar number is used in scaling to alter a time series' global magnitude or intensity. Scaling involves multiplying the scaling parameter α by the total time series. Equation (2) represents mathematical notation of scaling.

$$\hat{X}_s = \alpha x_1, \ldots, \alpha x_t, \ldots, \alpha x_T, \tag{2}$$

The scaling parameter α is selected via a Gaussian distribution (N) $\alpha \in N(0.5, \sigma^2)$ with σ 1.5. Figure 6 demonstrates the actual and transformed data after applying scaling on CogAge atomic (bending) activity.

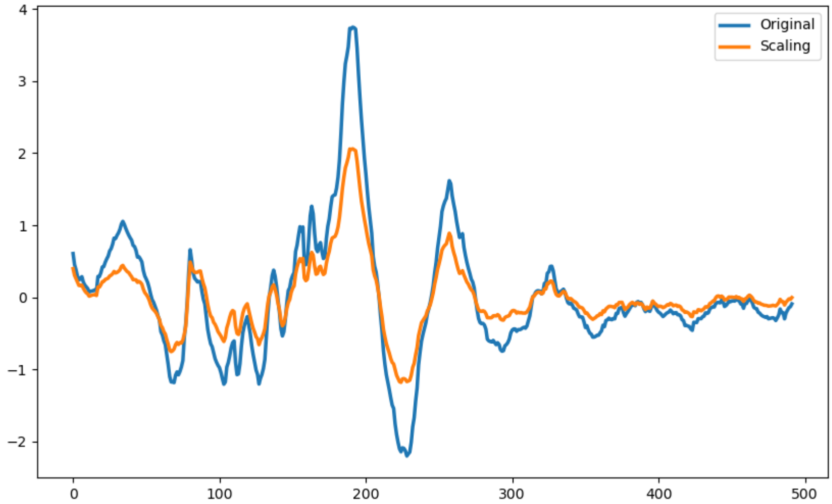

Figure 6. A visual representation of actual data and transformed data after applying scaling on a CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

3.1.2. Augmentation Based on TDTs

Transformations from the magnitude domain to the time domain are comparable with the exception that the transformation occurs along the time axis. Alternatively stated, the time series elements are displaced to distinct time steps from their initial position. In the following, we explain different methods of time domain transformations.

- **Time warping:** This refers to the process of altering a pattern in the temporal dimension. This task accomplished by employing a seamless distortion trajectory [62,85]. Equation (3) represents the mathematical notation of time warping.

$$\hat{X}_t = x_t(1), \ldots, x_t(t), \ldots, x_t(T) \tag{3}$$

Here, $\tau(\cdot)$ denotes a warping function that adjusts the time steps based on a smooth curve. The curve's smoothness is dictated by a cubic spline $S(u)$ with knots $u = u_1, \ldots, u_i, \ldots, u_I$. The knot heights u_i are determined from $N(0.5, 1.5)$. This transformation manipulates the time axis by compressing or expanding it at various points in the time series, which introduces diversity and enhances the dataset. Figure 7 demonstrates the actual and transformed data after applying time warping on CogAge atomic (bending) activity.

- **Linear Interpolation:** This calculates new values by fitting a straight line between neighboring data points. The interpolated value $\hat{X}(t)$ between two existing data points $X(t_i)$ and $X(t_{i+1})$ at any time t can be calculated by the Equation (4).

$$\hat{X_l i} = X(t_i) + (X(t_{i+1}) - X(t_i)) \times \frac{t - t_i}{t_{i+1} - t_i} \tag{4}$$

where $X(t_i)$ and $X(t_{i+1})$ represent the values of sequences at time t_i and t_{i+1}, respectively. Figure 8 demonstrates the actual and transformed data after applying linear interpolation on CogAge atomic (bending) activity.

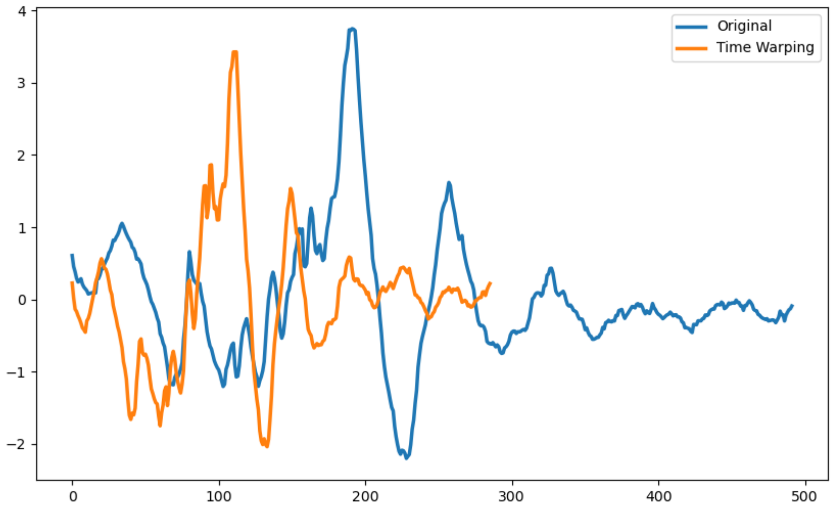

Figure 7. A visual representation of actual data and transformed data after applying time warping on CogAge atomic (bending) activity. The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

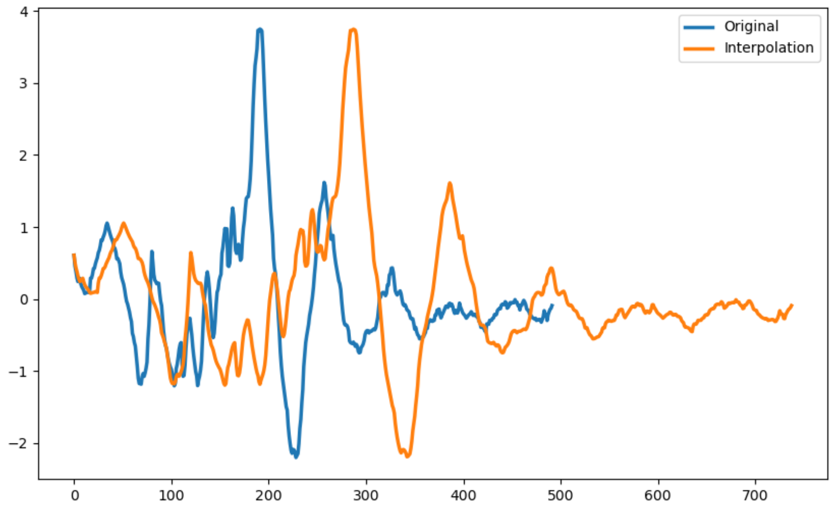

Figure 8. A visual representation of actual data and transformed data after applying linear interpolation on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

- **Exponential Moving Median Smoothing:** This provides smooth data by exponentially diminishing weights. To be resistant against outliers, it computes a weighted median rather than a weighted average. Mathematically, it can be expressed by Equation (5).

$$\hat{X}_e = [\text{med}(x(t)), \text{med}(x(t+1)), \ldots, \text{med}(x(t+W-1))] \quad (5)$$

where $x(t)$ represents the input data sequence. *med* represents the median function and W represents window size. Figure 9 demonstrates the actual and transformed data after applying smoothing on CogAge atomic (bending) activity.

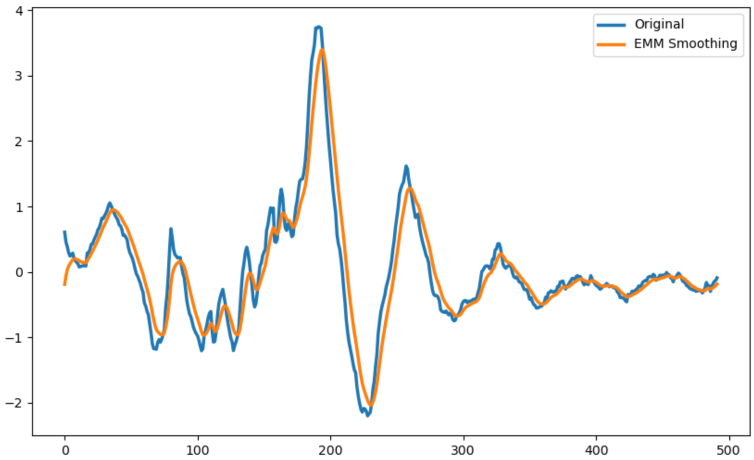

Figure 9. A visual representation of actual data and transformed data after applying exponential moving median smoothing on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

- **Channel Permutation:** This shuffles the channels (or features) of the complete sequence without changing the values inside each channel. Mathematically, it is represented by Equation (6).

$$\hat{X_p} = x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)} \tag{6}$$

X is the rearranged version of the original data, where π is a function that changes the order of the channels. This transformation preserves the chronological order of the data but adds diversity by reorganising the channels. Figure 10 demonstrates the actual and transformed data after applying channel permutation on CogAge atomic (bending) activity.

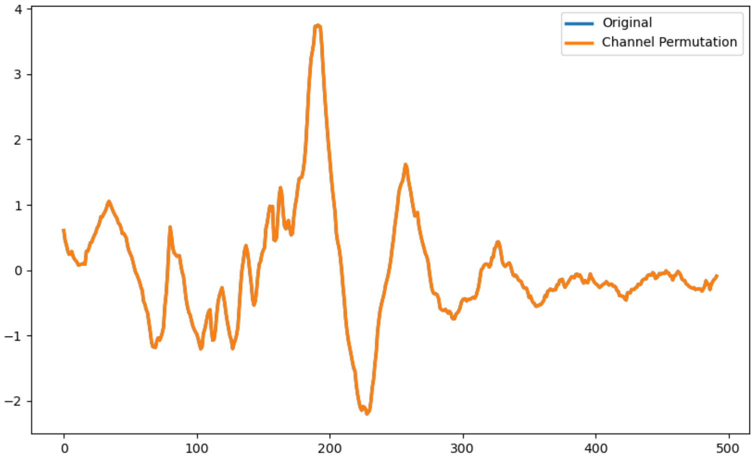

Figure 10. A visual representation of actual data and transformed data after applying channel permutation on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

- **Rolling Window Averaging:** This helps to smooth or denoise the data while maintaining its underlying structure. This method entails utilising a moving average process on the time series data by employing a sliding window. Equation (7) represents the mathematical notation of it.

$$\hat{X_a} = [f(x(t)), f(x(t+1)), \ldots, f(x(t+W-1))] \tag{7}$$

where $\hat{X_a}$ indicates the outcome of applying the rolling window operation on the time series $x(t)$ with a window size 10. $f(x'(t) = \frac{1}{W}\sum_{i=t}^{t+W-1} x(i))$ symbolises the window function applied to each subset of the time series data, t represents the current time index and the rolling window progresses along the time axis. The rolling window procedure produces a new time series by calculating each value using a window function on consecutive portions of the original time series data. Figure 11 demonstrates the actual and transformed data after applying averaging smoothing on CogAge atomic (bending) activity.

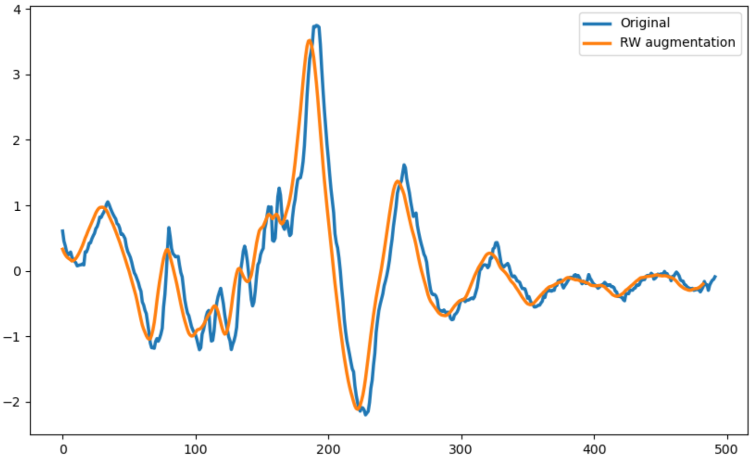

Figure 11. A visual representation of actual data and transformed data after applying rolling window averaging on CogAge atomic dataset (bending activity). The X-axis represents time in milliseconds, while the y-axis represents data sequences for bending activity.

3.1.3. Mixing Pattern

Pattern mixing is the process of combining two or more patterns to create new ones. For random transformations, it is assumed that the transformation results are representative of the dataset. Not every transformation, however, is appropriate for every dataset. Pattern mixing has the advantage of not making this same assumption. Pattern mixing, on the other hand, presupposes that similar patterns can be blended and produce good outcomes [74].

- **Sub Averaging:** This process involves the averaging of two patterns to produce a unique pattern. This method includes averaging the temporal values of two sequences belonging to the same class to generate a novel time series pattern, similar to the mixup approach [33]. We integrate different subjects within the same class. Mathematically, it is formulated as in Equation (8)

$$\hat{X_s a} = x'_1, x'_2, \dots, x'_n \tag{8}$$

The new time series pattern $\hat{X_s a}$ is calculated by taking the average of the corresponding values in $X_1 = [x_{11}, x_{12}, \dots, x_{1n}]$ and $X_2 = [x_{21}, x_{22}, \dots, x_{2n}]$. This method increases the diversity of the dataset and can help the generalisation of the model by generating a wide range of training samples. Figure 12 demonstrates the actual and transformed data after applying sub averaging on CogAge atomic (bending) activity.
- **Sub Cutmix:** This technique involves the random replacement of segments from two different sequences belonging to different subjects. It improves dataset variety by combining information from various time series, which can strengthen the resilience and generalisation capabilities of time series models. Mathematically, it can be expressed by Equation (9)

$$\hat{X_s c} = x_{11}, x_{12}, \dots, x_{1i}, x'_2, x'_3, \dots, x'_k, x_{1k+1}, \dots, x_{1n} \tag{9}$$

where $x'_2, x'_3, \dots, x'_k$ denote a segment that is substituted from $X_2 = [x_{21}, x_{22}, \dots, x_{2n}]$ in $X_1 = [x_{11}, x_{12}, \dots, x_{1n}]$. The indices i and k are randomly chosen to determine the start and finish positions of the segment substitution. Figure 13 demonstrates the actual and transformed data after applying transformation on CogAge atomic (bending) activity.

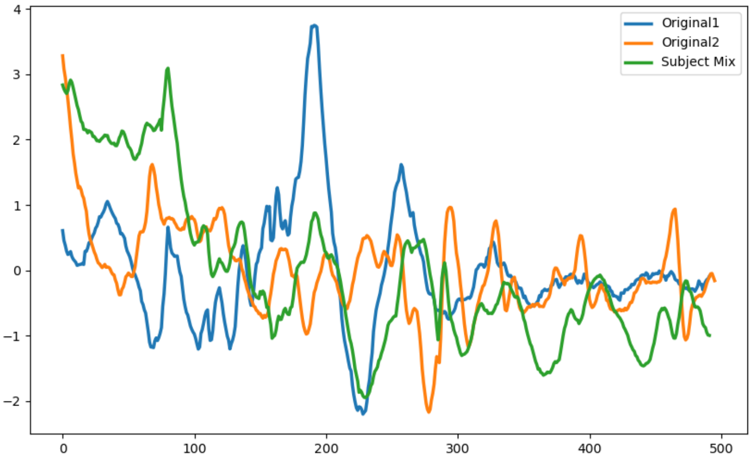

Figure 12. A visual representation of actual data and transformed data after applying sub averaging on the CogAge atomic dataset (bending activity). The X-axis represents time in milliseconds, while the y-axis represents data sequences for bending activity.

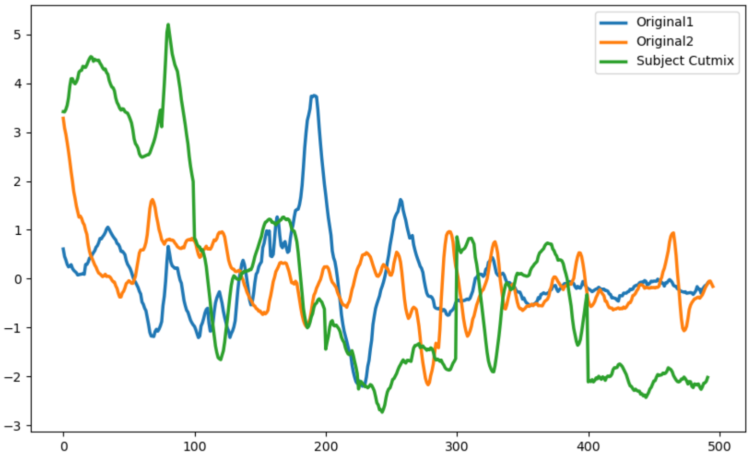

Figure 13. A visual representation of actual data and transformed data after applying sub cutmix on CogAge atomic dataset (bending activity). The X-axis represents time in milliseconds, while the y-axis represents data sequences for bending activity.

- **AugMix**: This uses three simultaneous augmentation chains with randomly selected augmentation operations. Three transformed sequences (smoothing, scaling, time warping) are created by consecutively applying these operations to the input sequence. The altered data are mixed with original data to create a new sequence. Incorporation of different transformation techniques directly into data increases the variability and robustness of models. Figure 14 demonstrates the actual and transformed data after applying AugMix on CogAge atomic (bending) activity.
- **Mixup**: This is used to combine two randomly selected time series to create new sequences. During data blending, the mixing factor λ sets the fraction of values from each sequence $(\lambda\ \epsilon\ [0,1])$ [32]. It is mathematically represented by Equation (10):

$$\hat{X_m}u = \lambda \cdot x_1 + (1 - \lambda) \cdot x_2 \tag{10}$$

Here, $\hat{X_m}u$ represents the augmented sequences generated by blending sequences $X_1 = [x_{11}, x_{12}, \ldots, x_{1n}]$ and $X_2 = [x_{21}, x_{22}, \ldots, x_{2n}]$ based on the mixing factor λ. The values for λ are typically drawn from a beta distribution. Figure 15 demonstrates the actual and transformed data after applying mixup on CogAge atomic (bending) activity.

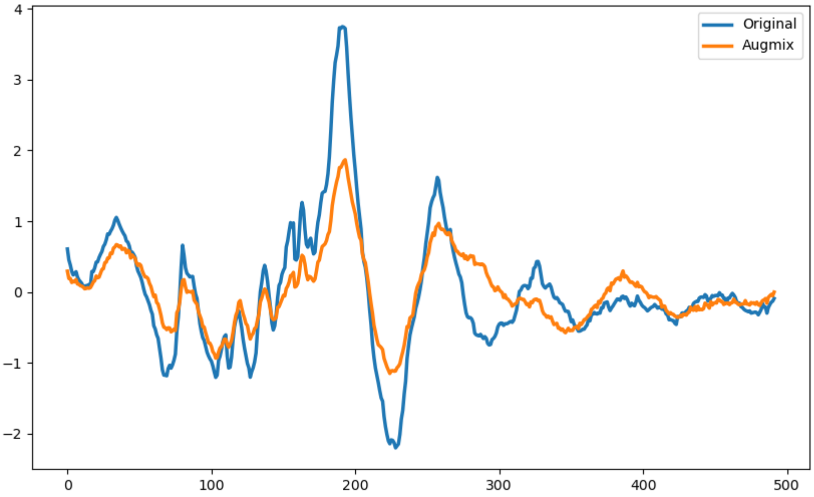

Figure 14. A visual representation of actual data and transformed data after applying AugMix on CogAge atomic dataset (bending activity). The X-axis represents time in milliseconds, while the y-axis represents data sequences for bending activity.

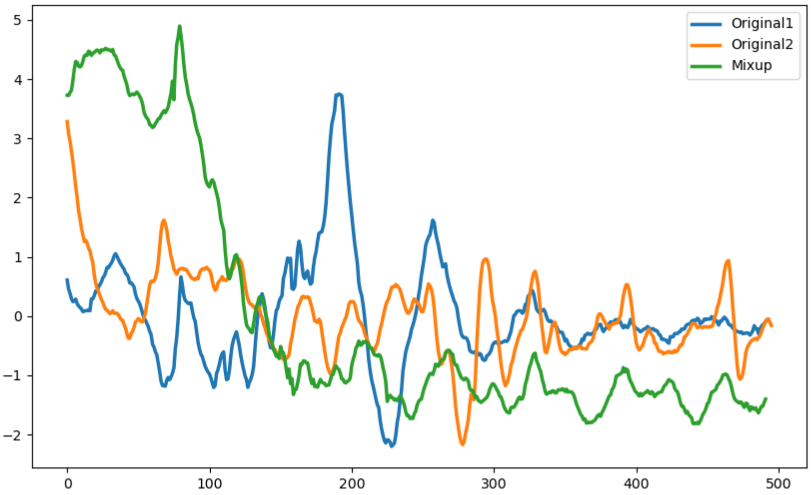

Figure 15. A visual representation of actual data and transformed data after applying mixup on CogAge atomic dataset (bending activity). The X-axis represents time in milliseconds, while the y-axis represents data sequences for bending activity.

- **Cutmix**: This technique substitutes arbitrarily shaped segments from one image for the other [75,76]. This can be stated mathematically by Equation (11).

$$X_{\hat{c}}m = \alpha \cdot X_1 + (1 - \alpha) \cdot X_2 \tag{11}$$

X_1 and X_2 are two initial data patterns. The new data pattern, $X_{\hat{c}}m$, is produced by combining X_1 and X_2 with a mixing coefficient of α. The amount of original patterns that are kept in the blended pattern is controlled by α which is empirically set to 0.2. By adding variability to the data, this method increases the diversity of datasets and may also strengthen the generalisation and robustness of the models. Figure 16 demonstrates the actual and transformed data after applying cutmix on CogAge atomic (bending) activity.

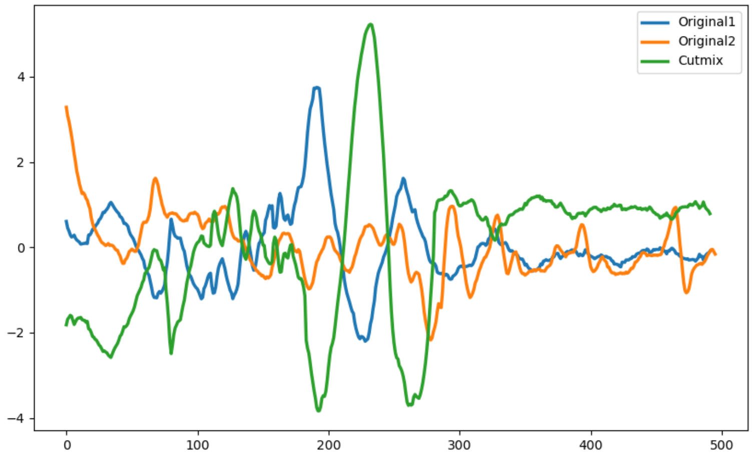

Figure 16. A visual representation of actual data and transformed data after applying cutmix on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

- **Hide and Seek:** This splits up sequences into a predetermined number of segments or intervals using the hide and seek strategy. Next, a random selection process is used to mask each segment with a specific probability, thus concealing its information. Random parts of the time series are eliminated by substituting the average of all the data points in the dataset for the masked segments. By replicating missing or noisy data, this approach increases variability and can improve the robustness of time series models. Figure 17 demonstrates the actual and transformed data after applying hide and seek on CogAge atomic (bending) activity.

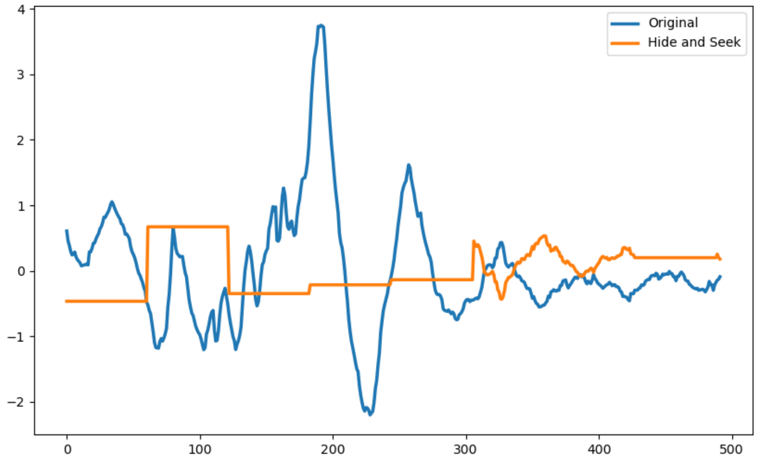

Figure 17. A visual representation of actual data and transformed data after applying hide and seek on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

3.1.4. Other Techniques

Apart from above-mentioned techniques, there are two other hybrid techniques that we employed for data augmentation, namely tsaug [86] and sequential transformation. These techniques use certain features from the above-mentioned categories of data augmentation and combine them with other techniques to give results. Data visualisation of these two techniques showing results of original and augmented data is given in Figures 18 and 19 respectively.

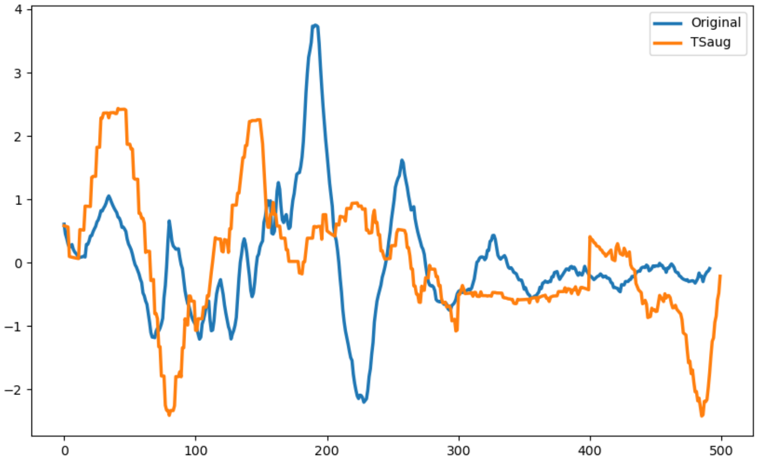

Figure 18. A visual representation of actual data and transformed data after applying tsaug on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

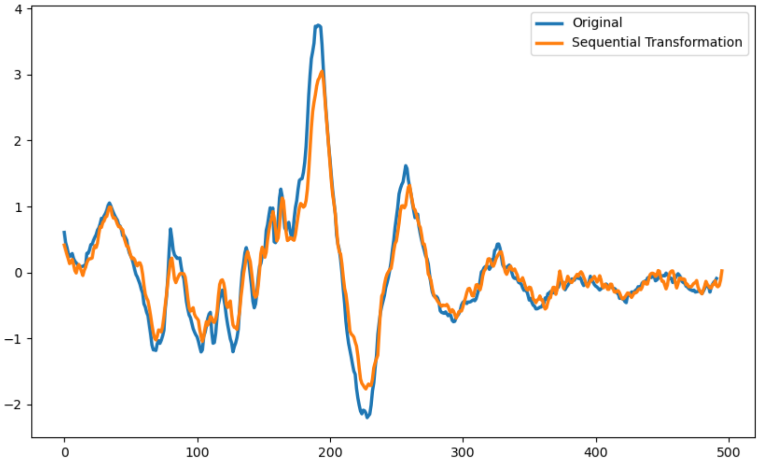

Figure 19. A visual representation of actual data and transformed data after applying sequential transformation on CogAge atomic dataset (bending activity). The *X*-axis represents time in milliseconds, while the *y*-axis represents data sequences for bending activity.

3.2. Model Architecture Design

We recall that the majority of the existing work does not effectively address the management of data from various modalities. Each sensing modality produces data at a distinct rate; for instance, smart glasses produce data at a rate of 20 Hz per second, whereas smart watches produce 100 Hz per second. However, employing deep learning models becomes futile once features have been derived from unprocessed data. We gravitated toward these models due to the fact that they generate features directly from unprocessed data. To cope up with this problem, implementation of an appropriate model for the data type at hand is needed. This research presents the MHyCoL network which consists of a multi-branch hybrid (CNN-LSTM) network, in which a distinct branch corresponds to each modality. These branches receive data of variable length, process them to produce features, concatenate these features at a subsequent stage and employ feature learning to perform classification.

3.2.1. Convolutional Neural Network

A popular deep learning model for handling organised grid-like data, such as digital images, is the CNN. The architecture of this model comprises several layers, namely convolutional layers, pooling layers and fully connected layers. CNNs employ convolutional operations to acquire hierarchical representations directly from the input data, facilitating efficient feature extraction and pattern identification. A typical convolutional block is composed of a convolutional layer, which is subsequently followed by a nonlinear activation function, such as the Rectified Linear Unit (ReLU) and a pooling layer for the purpose of down-sampling. The functioning of the CNN block can be mathematically represented by Equation (12)

$$z_{i,j}^{(l)} = \sum_{k=0}^{K-1}\sum_{s=0}^{S-1}\sum_{t=0}^{T-1} w_{k,s,t}^{(l)} \cdot x_{i+s,j+t,k}^{(l-1)} + b_i^{(l)} \tag{12}$$

where $z_{i,j}^{(l)}$ is the output feature map at position (i,j) in layer l. $x_{i+s,j+t,k}^{(l-1)}$ represents the input feature map at position $(i+s, j+t, k)$ in layer $(l-1)$. The convolutional kernel (weight) at position (k,s,t) in layer l is $w_{k,s,t}^{(l)}$. $b_i^{(l)}$ is the bias term for the i-th output channel in layer l. K, S, and T are the dimensions of the kernel.

$$f(x) = \max(0, x) \tag{13}$$

Equation (13) represents an activation function which in most of the cases is ReLU. The ReLU introduces nonlinearity to the network, facilitating the learning of intricate patterns.

$$y_{i,j}^{(l)} = \max_{s,t}\left(x_{Si,Sj}^{(l)}\right) \tag{14}$$

where $y_{i,j}$ represents the output feature map at position (i,j) in layer l, $x_{Si,Sj}$ denotes the region of the input feature map covered by the pooling window at position (Si, Sj) in layer l. $\max_{s,t}$ performs the computation of the highest value across the spatial dimensions s and t within the pooling window. The utilisation of pooling layers is employed to downsample the feature maps, hence diminishing the spatial dimensions of the data while simultaneously retaining significant characteristics. Figure 20 represents CNN architecture comprising of convolution, pooling and dense layer.

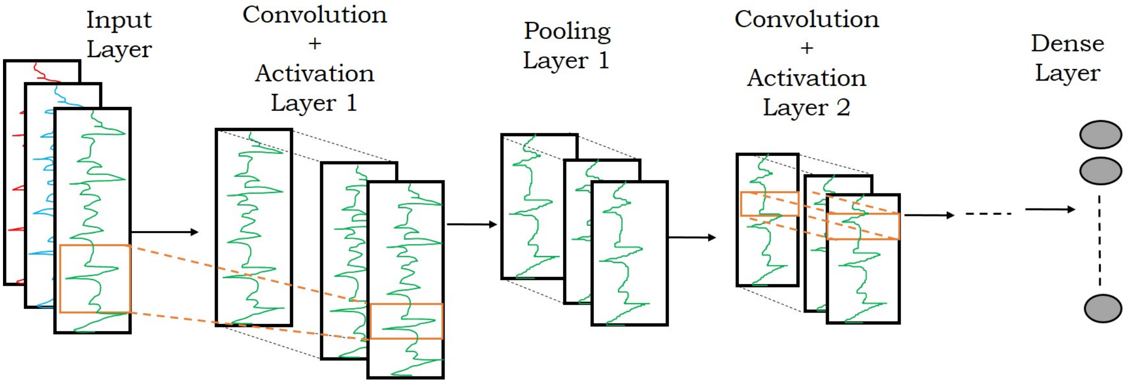
Input Layer
Convolution + Activation Layer 1
Pooling Layer 1
Convolution + Activation Layer 2
Dense Layer

Figure 20. A visual representation showcasing CNN architecture comprising of convolution, pooling and dense layer.

3.2.2. Long Short-Term Memory

LSTM is a variant of recurrent neural networks (RNNs) that is proficient in handling temporal data. Time series data analysis usually explores long-term dependencies and patterns. In contrast to conventional RNNs, LSTM models include memory cells capable of retaining information for prolonged durations. This feature serves to address the issue of disappearing gradients that might arise during the back-propagation process. In order to retain information from the prior time stamp and the present one, the system utilised input, forget and output gates [87]. The input gate manages the state of the cell by utilising data from the current time stamp and the reserved information stored in the memory cell. The forget gate is responsible for regulating the quantity of data that must be eliminated from the memory cell. The sigmoid function is employed to ascertain the data that should be eliminated if it is no longer pertinent. The function of the output gate is to regulate the selection of information from the cell state that will generate an output at the present time stamp. Figure 21 showcases the internal arrangement of the LSTM architecture.

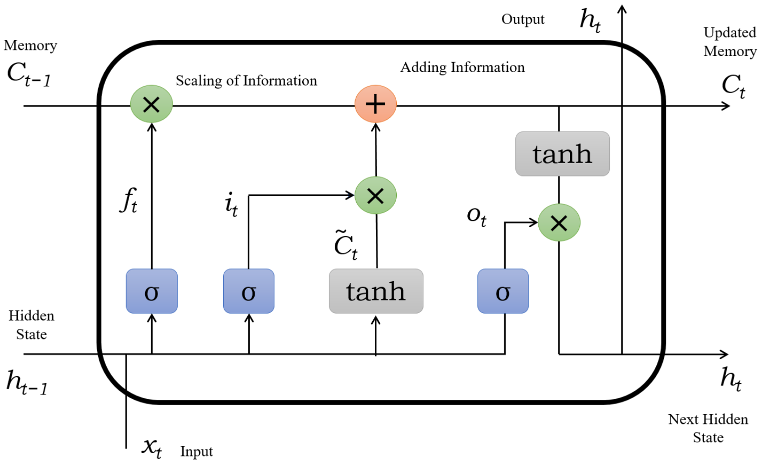

Figure 21. A visual representation showcasing the internal arrangement of the LSTM architecture.

Let x_t denote an input at time stamp t, h_{t-1} represent the hidden state of the preceding time stamp $t-1$, W be the weight matrix and b be the bias vectors for gates. Furthermore, the output gate is denoted as o, the input gate is represented by i and the forget gate denoted by f. The memory cell combines the information from the previous memory cell state C_{t-1} by multiplying the output of the forget gate ft with the new candidate information $i_t \odot \widetilde{C}_t$. The functioning of the gates can be mathematically represented by Equations (15)–(20):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (15)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (16)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (17)$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (18)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \quad (19)$$

$$h_t = o_t \odot \tanh(C_t) \quad (20)$$

where C_t denotes the state of the cell at time t, $\odot$ represents the point-wise multiplication operations and represents the activation functions used to compress the cell's information. The LSTM's output for the current time stamp t is determined by applying a hyperbolic tangent function tanh to the output gate o_t, which corresponds to the memory cell state C_t.

3.2.3. Multi-Branch Hybrid Conv-LSTM (MHyCoL)

Many individual models in the field of deep learning have been suggested in previous studies to extract a suitable feature representation from temporal sensory data. However, these models are restricted to encoding only one aspect of the data and are not capable of capturing the intricate relationships between the patterns. This paper introduces an ensemble model that effectively captures intricate patterns and interdependencies in temporal data. A powerful approach in machine learning involves combining multiple models, leveraging their individual strengths to create a more robust and efficient solution. To this end, we proposed an ensemble MHyCoL network to classify human activities of daily living using multimodal data of different wearable smart devices. The proposed ensemble MHyCoL model utilises a Hybrid (ConvLSTM) branch network to recognise human activities. This cutting-edge architecture consists of two unique models inside the paradigm. The primary model is around the utilisation of a branched CNN to model time series multimodal sensory data. This CNN operates simultaneously with changeable input. Each CNN branch in the model corresponds to a unique sensor modality having different frequencies. The CNN comprises of two convolutional layers utilising 32 and 16 filters, respectively, with a kernel size of 3 and l2 regulariser and activation function as 'relu' followed by two 1D pooling layers (Max pooling, pool size 2). The output of this layer is fed to a flatten layer followed by a dense layer with l2 regulariser and 'relu' activation for the computation of spatial characteristics. This layer is followed by a dropout layer with 50% dropout units. Subsequently, the combined spatial characteristics from all the CNN branches are inputted into the LSTM model. The LSTM model, specifically developed to capture temporal characteristics, is comprised of two layers. These layers specialise in acquiring sequential knowledge by utilising 128 and 64 memory units, enabling the storing and retrieval of information across sequential data. The LSTM layers utilise the 'relu' activation function. The softmax activation function is used in the output layer for multi-class classification. Figure 22 represents the architecture of proposed ensemble MHyCoL model.

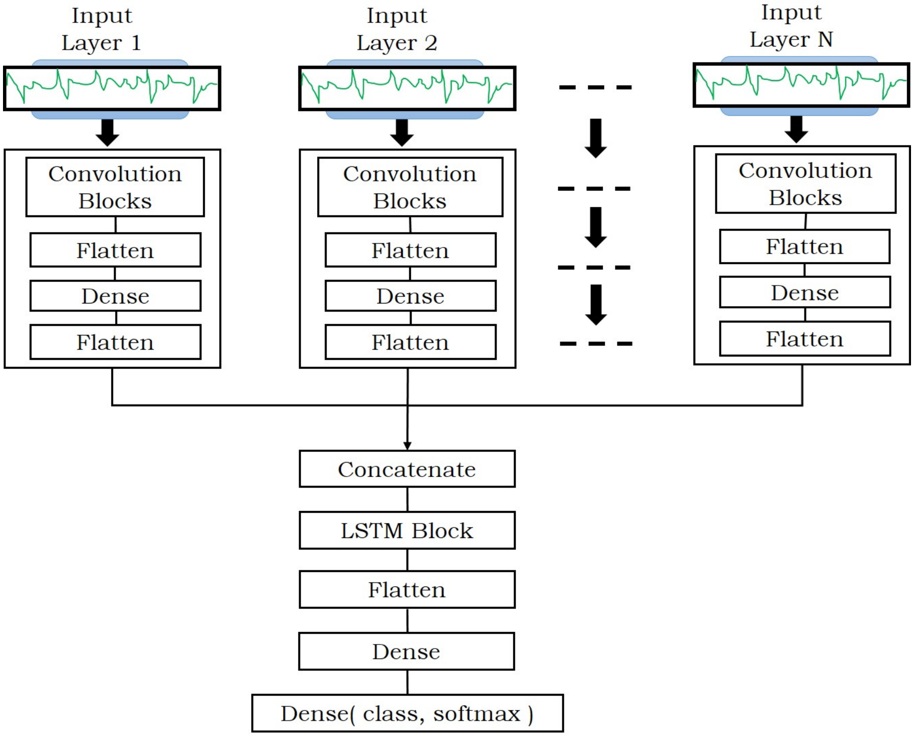

Figure 22. A visual representation showcasing the architecture of the proposed ensemble model.

4. Experiments and Results

4.1. Dataset Description

The performance of the proposed ensemble MHyCoL model was evaluated using two widely recognised publicly accessible datasets, namely CogAge [34] and UniMiB-SHAR [35]. The subsequent section provides a concise overview of each dataset.

4.1.1. CogAge

We employed the CogAge dataset in our experiments. The dataset is divided into two parts: CogAge-atomic and CogAge-composite [34]. The dataset was acquired from IMUs of smartphones, smartwatches and smartglasses. Each data instance is made up of nine sensor modalities, each with three sensor channels (x, y and z). The following sensor modalities are used:

1. Smartphone Accelerometer (sp-acc);
2. Smartphone Gyroscope (sp-gyro);
3. Smartphone Gravity (sp-grav);
4. Smartphone Linear Accelerometer (sp-linAcc);
5. Smartphone Magnetometer (sp-magn);
6. Smartwatch Accelerometer (sw-acc);
7. Smartwatch Gyroscope (sw-gyro);
8. Smartglasses Accelerometer (sg-acc);
9. Smartglasses Gyroscope (sg-gyro).

The CogAge-atomic dataset is divided into two further categories of short-term activities: postural and behavioural activities. Posture actions, such as standing, sitting and walking, indicate a subject's state. Behavioural activities, such as drinking, sweeping the floor and opening the door, show the task that a person is completing. Table 3 presents the summary of the CogAge dataset activity type and classes.

Table 3. Summary of CogAge dataset activity type and classes.

CogAge Dataset	**Activity Type**	**Classes**
Atomic	Postural	Standing, Sitting, Lying, Squatting, Walking, Bending
	Behavioural	Sit down, Stand up, Lie down, Get up, Squat down, Stand up from squatting, Open door, Close door, Open drawer, Close drawer, Open small box, Close small box, Open big box, Close big box, Open lid by rotation, Close lid by rotation, Open other lid, Close other lid, Open bag, Take from floor, Put on floor, Bring, Put on high position, Take from high position, Take out, Eat small thing, Drink, Scoop and put, Plug in, Unplug, Rotate, Throw out, Hang, Unhang, Wear jacket, Take off jacket, Read, Write, Type, Talk using telephone, Touch smartphone screen, Open tap water, Close tap water, Put from tap water, Put from bottle, Throw out water, Gargle, Rub hands, Dry off hands by shake, Dry off hands, Press from top, Press by grasp, Press switch/button, Clean surface, Clean floor.
Composite		Brushing Teeth, Cleaning Room, Handling Medication, Preparing Food, Styling Hair, Using Phone, Washing Hands

For each atomic activity instance, data were gathered for five seconds. However, due to data-transmission problems, not all channels must be exactly five seconds long. The dataset was collected by eight participants and contains 9029 occurrences of 61 atomic activities, 886 of which are state activities while the remaining 8143 are behavioural activities. Figure 23 shows a visual representation of the distribution of samples across various atomic activities in the CogAge dataset.

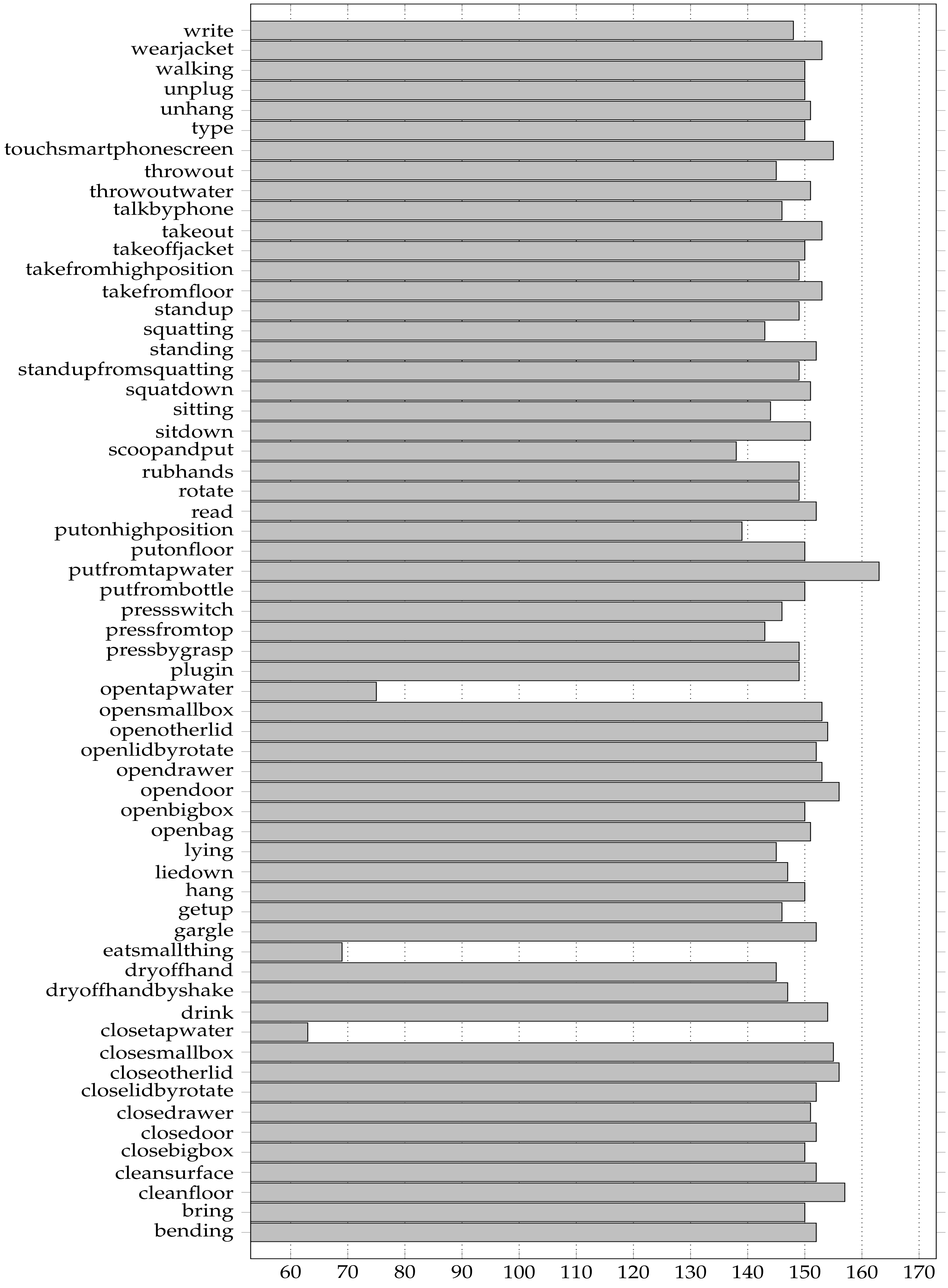

Figure 23. A visual representation of the distribution of samples across various atomic activities in the CogAge dataset. The *X*-axis represents example counts for each activity, while the *Y*-axis represents activities.

In contrast, the CogAge-composite dataset includes data on composite activities, where a participant engages in many tasks of daily life such as brushing teeth, cleaning a room and preparing food. The duration of each composite activity fluctuates in accordance with natural circumstances. The dataset was obtained from six individuals and comprises about 900 occurrences of seven composite activities. Figure 24 shows a visual representation of the distribution of samples across various composite activities in the CogAge dataset.

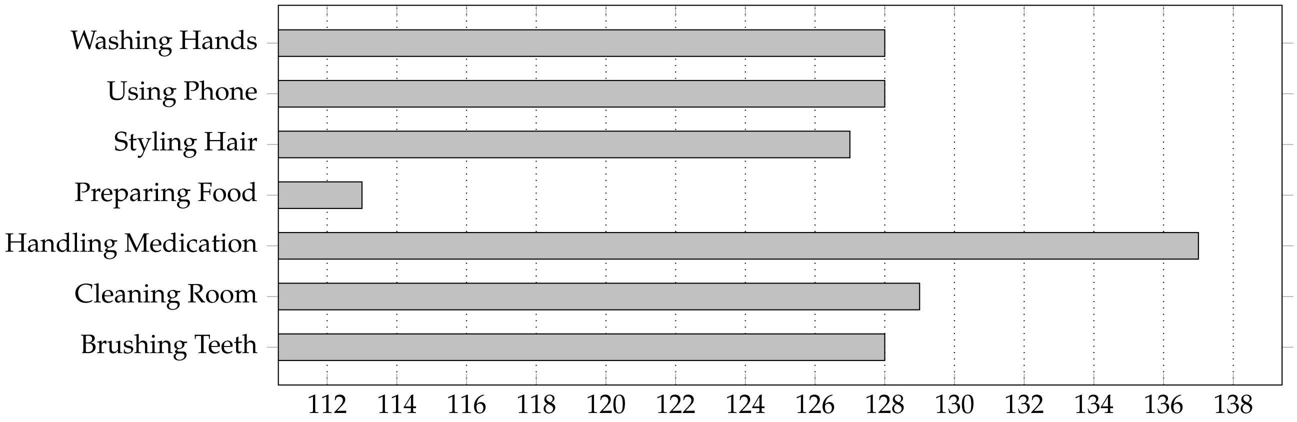

Figure 24. A visual representation of the distribution of samples across various composite activities in the CogAge dataset. The *X*-axis represents example counts for each activity, while the *Y*-axis represents activities.

4.1.2. UniMiBSHAR

The UniMiB-SHAR dataset contains information from 30 people (6 men and 24 women), captured with the 3D accelerometer of a Samsung Galaxy Nexus I9250 smartphone collected at 50 Hz. The dataset was further divided into eight "falling" actions and nine ADLs among the seventeen classes. The smartphone was carried out twice or six times for every activity, depending on which pocket it is in (left or right). This dataset is balanced, even though three ADL classes are more represented than the others. It also does not contain a null class. Table 4 represents the classification of UniMiB-SHAR data according to activity type and classes.

Table 4. Summary of UniMiB-SHAR data classification according to activity type and classes.

Activity Type	Classes
Fall	Falling forward, Falling right, Falling backward, Falling left, Hitting obstacle, Falling with protection strategies, Falling backward sitting on chair, Syncope
ADLs	Standing up from sitting, Standing up from lying, Walking, Running, Going upstairs, Jumping, Going downstairs, Lying down from standing, Sitting down

Figure 25 represents the breakdown of various activities recorded in the UniMiB-SHAR dataset. The activities were performed over 3 s with a set length 151. The dataset has 11,771 example points in total.

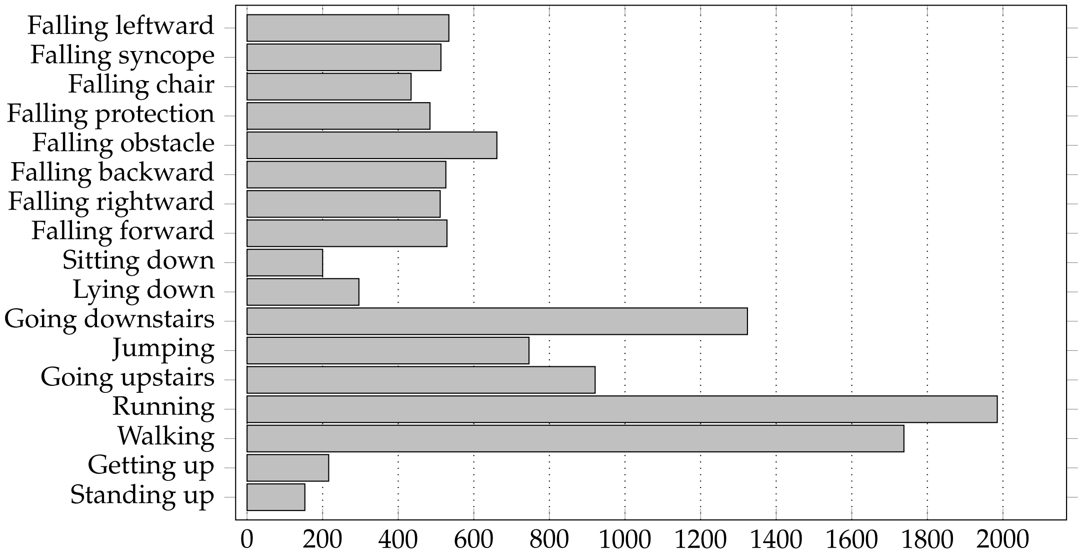

Figure 25. A visual representation of the distribution of samples across various composite activities in the UniMiB-SHAR dataset. The X-axis represents example counts for each activity, while the Y-axis represents activities.

4.2. Pre-Processing

All the datasets were pre-processed to make them suitable for the proposed model. While performing pre-processing on the CogAge atomic activity dataset, due to data-transmission problems related to unavailability of exactly five seconds of data instance, we opted to use the first four seconds of each data instance. The reason for selecting four seconds was that this was the size available for all data instances contrary to five seconds. CogAge composite activity data were synchronised first to keep all modality data in the same time because every modality starts producing data at a different time. We align the data for each modality based on the latest start time and earliest end time among all. Further, the data rate varies for each modality. For instance, smart glasses produce data at a rate of 20 Hz per second, smart watches produce data at a rate of 100 Hz per second and smart phones generate data at a rate of 200 Hz per second. We have meticulously separated each activity into data non-overlapping windows of five seconds. The UniMiB-SHAR dataset was provided in the window of three seconds, so we followed the same duration.

4.3. Datasetting

In order to assess the model's ability to generalise the CogAge dataset, we allocated half the data for training and the other half for testing. However, for the UniMibShar dataset we applied two different settings. The first setting used the initial 20 subjects for training and the last 10 subjects for testing. In the second setting, the entire dataset was distributed in a 60–40 ratio. Since our experiments were twofold, we performed data augmentation using fifteen different techniques resulting in double the amount of data in every case. The same distribution was followed after data augmentation.

4.4. Experimental Setup

The experimental evaluation is carried out on a 13th Gen Intel (R) Core (TM) i9-13900K 3.00 GHz processor, 64 GB RAM running on Windows 10 operating system, HP, Lahore, Pakistan. Tables 5 and 6 show the experimental settings for both UniMiB-SHAR and CogAge datasets.

Table 5. Summary of experiments carried out on UniMiB-SHAR dataset.

Experiment	Description
AF-17-1	This experiment involves classifying 17 different activities, including falls and activities of daily living (ADLs), with a data distribution of 20–10.
AF-17-2	This experiment falls under the classification of 17 activities, encompassing both fall and ADL classes, with a data distribution of 60–40.
ADL-9	This task evaluates the performance of recognising activity sequences from 9 ADL classes, with a data distribution of 60–40.
F-8	This task evaluates the performance of recognising activity sequences from 8 different Fall classes, using a data distribution of 60–40.
A-17	This experiment was conducted on a total of 17 activities. The testing was conducted by dividing into two sets: 8 Fall classes (AF8) and 9 ADL classes (AF9), with a 60–40 distribution.

Table 6. Summary of experiments carried out on CogAge dataset.

Experiment	Description
CC-7	This experiment falls under the classification of 7 composite activities with an equal distribution of subject data.
CA-61	This experiment falls under the classification of 61 atomic activities.
CAB-55	This task evaluates the performance of recognising atomic activity sequences across 55 different behaviour classes.
CAP-6	This task evaluates the performance of recognising atomic activity sequences across 6 different posture classes.
CA-2	This experiment was conducted on 61 atomic activities and subsequently tested on 6 posture classes (CA-6) and 55 behaviour classes (CA-55) individually.

The model is trained for 200 epochs for the CogAge dataset and 300 epochs for the UniMiB-SHAR dataset. While compiling the model, the optimiser is configured as ‘adam’ and the loss function ‘sparse categorical crossentropy’ is employed, indicating a meticulous and advanced approach in the construction of the deep learning model. Mathematically, it can be described as in Equation (21):

$$L(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) \tag{21}$$

where $L(y, \hat{y})$ is the sparse categorical cross-entropy loss. The number of samples denoted by n. y_i is the true class label for sample i, whereas $\hat{y}_i$ represents the predicted probability assigned to the true class label for sample i.

4.5. Results and Discussion

This section presents the results of our experiments discussed in Section 4.4 using our proposed MHyCoL with fifteen different augmentation techniques in order to identify the most suitable technique for time series multi modal sensory data. Furthermore, to show how the overfitting problem is resolved with data augmentation the training and validation accuracy before and after augmentation is depicted in Figures 26 and 27.

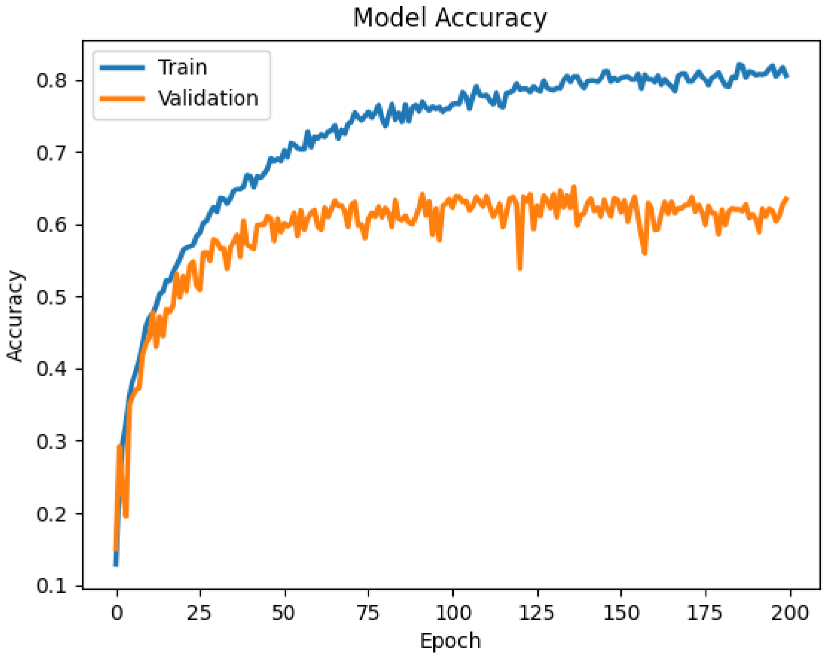

Figure 26. Training and validation accuracy graph of actual data for CogAge atomic activities.

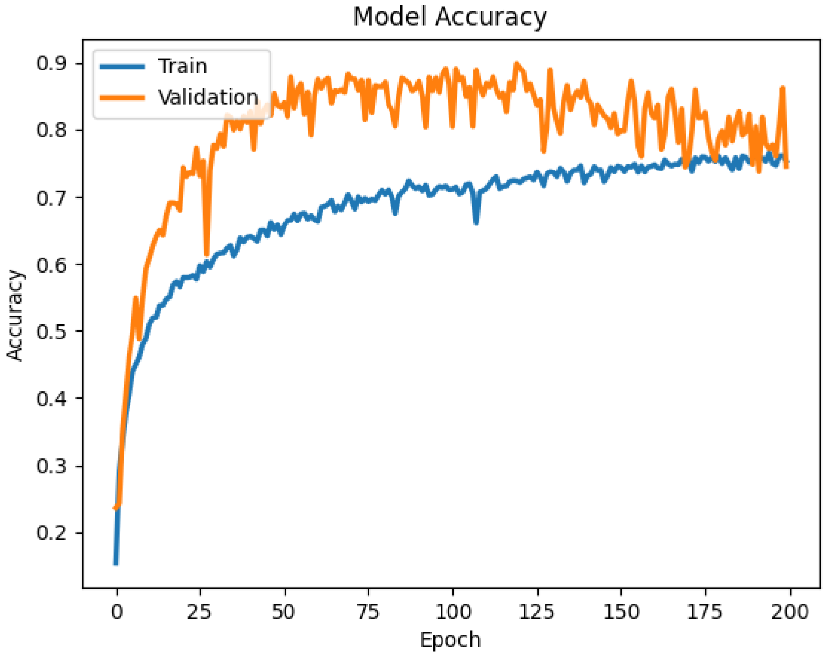

Figure 27. Training and validation accuracy graph of augmented data for CogAge atomic activities.

Table 7 shows the performance of the MHyCoL on CogAge datasets while employing different data-augmentation techniques discussed in Section 3.1. This way, we can identify which technique is better for what kinds of activities of daily living (ADLs). Using our actual non-augmented data, our model reported considerably low values of accuracies for all types of ADLs. For CAP-6 data, interpolation and median smoothing produced the best augmentation results with an accuracy of 99.75%. For CAB-55 data, time warping gave the best results with an accuracy of 92.5%. For the CA-61 dataset, time warping gave the best results with an accuracy of 84.73%. For the CC-7 dataset, scaling showed the best results, giving an accuracy of 84.12%.

Table 7. Summary of results from various data-augmentation techniques on CogAge dataset following the experimental setting of Table 6.

Augmentation Techniques	CAP-6	CAB-55	CA-61	CC-7
Actual	92.71	62.45	55.04	79.29
Interpolation	**99.75**	71.53	76.09	82.43
Jittering	96.84	74.66	73.34	80.65
Scaling	99.51	86.58	80.68	**84.12**
Median Smoothing	**99.75**	70.78	76.32	81.28
Rowling Mean Smoothing	93.8	62.51	60.58	76.19
Time Warping	98.54	**92.5**	**84.73**	82.16
AugMix	91.05	64.04	60.4	73.2
Cutmix	91.29	57.53	56.97	79.52
Hide and Seek	89.83	57.01	51.43	75.24
Mixup	87.89	59.43	63.18	78.87
Sequential Transformation	90.56	62.11	63.42	78.23
TSaug	91.29	50.87	51.26	80.38
Subject Mix	85.67	63.85	62.74	78.34
Subject Cutmix	23.72	38.23	40.88	63.53

Table 8 shows the performance of the MHyCoL on the UniMiB-SHAR dataset while employing different data-augmentation techniques discussed earlier. Using our actual non-augmented data, our model reported considerably low values of accuracies for fall and ADLs. For AF-17-1 data, median smoothing produced the best augmentation results with an accuracy of 72.41%. For AF-17-2 data, jittering and scaling gave the best results with an accuracy of 92.% and 91.63%, respectively. For the ADL-9 dataset, jittering and scaling produced the best results with an accuracy of 89.44% and 88.61%, respectively. Similarly, for the F-8 dataset, jittering showed the best results, giving an accuracy of 98.89%.

Table 8. Summary of results from various data-augmentation techniques on the UniMiB-SHAR dataset following the experimental setting of Table 5.

Augmentation Techniques	AF-17-1	AF-17-2	ADL-9	F-8
Actual	68.64	84.47	67.33	96.71
Interpolation	69.27	80.79	63.83	93.17
Jittering	70.71	**92**	**89.44**	**98.89**
Scaling	71.31	**91.63**	**88.61**	**98.81**
Median Smoothing	**72.41**	88.1	81.53	97.61
Rowling Mean Smoothing	70.69	86.6	84.23	96.05
Time Warping	71.3	88.01	75.32	**97.81**
AugMix	71.56	85.97	86.02	93.14
Cutmix	70.02	89.17	78.78	97.64
Hide and Seek	70.59	74.84	71.82	86.11
Mixup	69.76	79.57	80.03	65.89
Sequential Transformation	68.71	79.86	64.49	87.95
TSaug	69.79	85.04	70	97.15
Subject Mix	68.36	89.22	80.28	96.69
Subject Cutmix	62.33	78.82	50.32	85.23

A thorough analysis of the results shows that TDTs and MDTs played an important role in reducing the overfitting of the model for both datasets. Time warping performed well for the CogAge dataset, while jittering performed well for the UniMiB-SHAR dataset. The results further show that the subject cutmix is the worst technique for the time series multi modal sensory data. In fact most of the mixing pattern techniques that are very useful for image data augmentation did not perform well for the time series multi modal sensory data.

Because of basic differences in data structure and characteristics, these image-augmentation methods are designed to work only with images and not with time series data. Time series data are not appropriate for random augmentations due to their sequential nature and intricate temporal connections. Unlike pictures, time series do not have features that are localised in space, which makes it hard to figure out what these modifications mean. Using picture-enhancement methods on time series data messes them up with temporal patterns and makes the model less accurate.

Similarly, tsaug, which was built exclusively for time series data, does not perform well with time series sensory data due to its distinctive characteristics and requirements. Sensory data often consist of complicated temporal patterns and small changes that general augmentation techniques may not fully capture. The semantic meaning and temporal dependencies that are essential for the processing of sensory data are not preserved by tsaug, which leads to a decrease in model performance. In addition, sequential transformation does not work well for time series sensory data because of how complicated the temporal patterns are. These sequential changes mess up small temporal trends and add noise to the data that makes it hard to see important information. The complicated temporal dependencies of sensory input could not be kept well.

Table 9 shows the results of ADL-9, F-8 and A-17 (AF8 and AF9) from the UniMiB-SHAR dataset. Experiments show that when Fall and ADL data are trained on the model independently, the model shows good results but when the model is trained on combined data a sudden decline in the accuracy of these activities can be witnessed. This could be due to conflicting patterns or features between the two types of data. Table 10 shows the results for posture (CAP-6) and behaviour (CAB-55) data from the CogAge atomic data when trained individually, and it also presents results when the model is trained on 61 activities (posture and behaviour combined) and tests for posture (CA-6) and behaviour (CA-55) separately. It can be seen clearly that when the model was trained on posture and behaviour data individually, it presented better results but when it was trained on combined data, posture results drastically decreased while behaviour results show improvement. This leads to another research gap with respect to learning these activities simultaneously. Similarly, composite activity results by any of these handcrafted augmentation techniques are not promising as compared to atomic activities. This creates another dimension to explore techniques specifically designed to handle composite activities.

Table 9. Overview of findings from experiments conducted on the A-17, ADL-9 and F-8 from the UniMiB-SHAR dataset.

Augmentation Techniques	AF9	AF8	ADL-9	F-8
Actual	67.16	94.61	67.33	96.71
Interpolation	61.29	91.89	63.83	93.17
Jittering	83.58	96.79	89.44	98.89
Scaling	80.86	97.36	88.61	98.81
Median Smoothing	73.96	96.16	81.53	97.61
Rowling Mean Smoothing	69.51	95.39	84.23	96.05
Time Warping	60.65	91.75	75.32	97.81
AugMix	75.18	92.11	86.02	93.14
Cutmix	77.1	96.34	78.78	97.64
Hide and Seek	61.84	81.44	71.82	86.11
Mixup	68.76	64.67	80.03	65.89
Sequential Transformation	60.67	91.82	64.49	87.95
TSaug	68.08	94.98	70	97.15
Subject Mix	77.74	96.05	80.28	96.69
Subject Cutmix	56.75	88.32	50.32	85.23

Table 10. Overview of findings from experiments conducted on the CAP-6, CAB-55 and CA-2 from CogAge dataset.

Augmentation Techniques	CAP-6	CAB-55	CA-6	CA-55
Actual	92.71	62.45	62.14	54.25
Interpolation	99.75	71.53	79.36	75.53
Jittering	96.84	74.66	75.24	73.13
Scaling	99.51	86.58	71.35	81.7
Median Smoothing	99.75	70.78	77.71	76.19
Rowling Mean Smoothing	93.8	62.51	63.62	59.14
Time Warping	98.54	92.5	77.42	85.53
AugMix	91.05	64.04	59.82	57.41
Cutmix	91.29	57.53	61.92	56.4
Hide and Seek	89.83	57.01	52.94	51.26
Mixup	87.89	59.43	65.32	62.94
Sequential Transformation	90.56	62.11	66.53	63.07
TSaug	91.29	50.87	50.32	55.48
Subject Mix	85.67	63.85	67.99	62.16
Subject Cutmix	23.72	38.23	39.32	37.88

4.6. Performance Comparison with Existing State-of-the-Art Techniques

We also evaluated the performance of the proposed ensemble model by comparing it to other recently developed state-of-the-art models. The evaluation of all existing techniques was conducted on the CogAge and UniMiB-SHAR datasets, ensuring that the training and testing instances were distributed in the same manner. The outcomes of the recognition process are condensed and shown in Table 11. The proposed algorithm demonstrated superior performance compared to existing techniques, with Transformer [88], Random Forest [39], Rank pooling + SVM [34], CNN-transfer [6] and GILE [89] models achieving the highest recognition scores. The exceptional performance of the proposed ensemble model confirms its supremacy in accurately identifying human activities.

Table 11. The proposed model's recognition accuracy compared to recent state-of-the-art techniques using the CogAge and UniMib-SHAR datasets. The symbol '–' signifies that the technique is either not applicable to this dataset or the authors have not disclosed the results. The highest scores are emphasised in bold.

Method	Year	CogAge Atomic	CogAge Composite	UniMiB-SHAR
Transformer [88]	2022	–	73.36%	–
Random Forest [39]	2021	–	79%	–
Rank pooling + SVM [34]	2020	–	68.65%	–
CNN-transfer [6]	2020	state—95.94%, behaviour—71.8%	–	–
GILE [89]	2021	–	–	70.31%
Fusion [3]	2018	–	–	74.66%
CNN [72]	2023	–	–	78.83%
HM + RF [90]	2022	–	–	80.27%
Proposed Model	2024	**state—98.54%, behaviour—92.5%**	**82.16%**	**88.01%**

5. Conclusions

This study introduces the MHyCoL network to recognise time series multimodal sensory activity sequences. In addition, we conducted a systematic evaluation of fifteen different random transformation based data-augmentation techniques used on time series multimodal sensory data to solve the inadequacy problem of labeled data. An extensive evaluation of ensemble models is performed on two well-known benchmark datasets: CogAge and UniMiB-SHAR. These techniques produced a 5% improvement in accuracy for composite activities and a significant 30% boost for atomic activities. The increase in time series sensory data poses distinct challenges and possibilities in improving model resilience and generalisation. Although typical image-augmentation techniques like cutmix and mixup may not be directly suitable, domain-specific approaches such as time domain transformations and magnitude domain transformations demonstrate potential. These techniques maintain important changes over time and patterns related to frequency, effectively dealing with the intricate features of sensory data. However, the efficacy of augmentation approaches relies on their capacity to uphold semantic significance and temporal inter-dependencies. In the future, we are going to fill the gap of the model learning multiple activities simultaneously. Furthermore, exploration of deep learning-based data-augmentation models for composite activities to handle their long term dependencies can be a substantial research area for the future.

Author Contributions: Conceptualisation, N.A., M.H.K. and M.A.N.; methodology, N.A. and M.H.K.; software, N.A. and M.A.N.; validation, N.A.; investigation, N.A., M.H.K. and M.A.N.; writing—original draft preparation, N.A.; writing—review and editing, M.H.K. and M.A.N.; supervision, M.H.K. and M.A.N.; project administration, M.H.K. and M.A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Higher Education Pakistan within the project "HumCare: Human Activity Analysis in Healthcare" (Grant Number: 15041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the datasets used in this study are publically available and their references are provided in the dataset section.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bisio, I.; Lavagetto, F.; Marchese, M.; Sciarrone, A. Smartphone-centric ambient assisted living platform for patients suffering from co-morbidities monitoring. *IEEE Commun. Mag.* **2015**, *53*, 34–41. [CrossRef]
2. Batool, S.; Khan, M.H.; Farid, M.S. An ensemble deep learning model for human activity analysis using wearable sensory data. *Appl. Soft Comput.* **2024**, *159*, 111599. [CrossRef]
3. Li, F.; Shirahama, K.; Nisar, M.A.; Köping, L.; Grzegorzek, M. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors* **2018**, *18*, 679. [CrossRef] [PubMed]
4. Logan, B.; Healey, J.; Philipose, M.; Tapia, E.M.; Intille, S. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the International Conference on Ubiquitous Computing*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 483–500.
5. Yang, R.; Wang, B. PACP: A position-independent activity recognition method using smartphone sensors. *Information* **2016**, *7*, 72. [CrossRef]
6. Li, F.; Shirahama, K.; Nisar, M.A.; Huang, X.; Grzegorzek, M. Deep transfer learning for time series data based on sensor modality classification. *Sensors* **2020**, *20*, 4271. [CrossRef]
7. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: Review, opportunities and challenges. *Brief. Bioinform.* **2018**, *19*, 1236–1246. [CrossRef]
8. Avati, A.; Jung, K.; Harman, S.; Downing, L.; Ng, A.; Shah, N.H. Improving palliative care with deep learning. *BMC Med. Inform. Decis. Mak.* **2018**, *18*, 55–64. [CrossRef]
9. Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [CrossRef]
10. Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [CrossRef]
11. Choi, K.; Yi, J.; Park, C.; Yoon, S. Deep learning for anomaly detection in time series data: Review, analysis, and guidelines. *IEEE Access* **2021**, *9*, 120043–120065. [CrossRef]
12. Khan, M.H.; Farid, M.S.; Grzegorzek, M. A nonlinear view transformations model for cross-view gait recognition. *Neurocomputing* **2020**, *402*, 100–111. [CrossRef]
13. Naaz, F.; Herle, A.; Channegowda, J.; Raj, A.; Lakshminarayanan, M. A generative adversarial network-based synthetic data augmentation technique for battery condition evaluation. *Int. J. Energy Res.* **2021**, *45*, 19120–19135. [CrossRef]
14. Khan, M.H.; Azam, H.; Farid, M.S. Automatic multi-gait recognition using pedestrian's spatiotemporal features. *J. Supercomput.* **2023**, *79*, 19254–19276. [CrossRef]
15. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]

16. Olson, M.; Wyner, A.; Berk, R. Modern neural networks generalise on small datasets. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 3623–3632.
17. Blagus, R.; Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinform.* **2013**, *14*, 106. [CrossRef] [PubMed]
18. Hasibi, R.; Shokri, M.; Dehghan, M. Augmentation scheme for dealing with imbalanced network traffic classification using deep learning. *arXiv* **2019**, arXiv:1901.00204.
19. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
20. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
21. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
24. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
26. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time series data augmentation for deep learning: A survey. *arXiv* **2020**, arXiv:2002.12478.
27. Fields, T.; Hsieh, G.; Chenou, J. Mitigating drift in time series data with noise augmentation. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; pp. 227–230.
28. Um, T.T.; Pfister, F.M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 216–220.
29. Nweke, H.F.; Teh, Y.W.; Al-Garadi, M.A.; Alo, U.R. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Syst. Appl.* **2018**, *105*, 233–261. [CrossRef]
30. Min, J.; Tu, J.; Xu, C.; Lukas, H.; Shin, S.; Yang, Y.; Solomon, S.A.; Mukasa, D.; Gao, W. Skin-interfaced wearable sweat sensors for precision medicine. *Chem. Rev.* **2023**, *123*, 5049–5138. [CrossRef] [PubMed]
31. Jaitly, N.; Hinton, G.E. Vocal tract length perturbation (VTLP) improves speech recognition. *Proc. ICML Workshop Deep. Learn. Audio Speech Lang.* **2013**, *117*, 21.
32. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
33. Inoue, H. Data augmentation by pairing samples for images classification. *arXiv* **2018**, arXiv:1801.02929.
34. Nisar, M.A.; Shirahama, K.; Li, F.; Huang, X.; Grzegorzek, M. Rank pooling approach for wearable sensor-based ADLs recognition. *Sensors* **2020**, *20*, 3463. [CrossRef] [PubMed]
35. Micucci, D.; Mobilio, M.; Napoletano, P. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Appl. Sci.* **2017**, *7*, 1101. [CrossRef]
36. Urwyler, P.; Rampa, L.; Stucki, R.; Büchler, M.; Müri, R.; Mosimann, U.P.; Nef, T. Recognition of activities of daily living in healthy subjects using two ad-hoc classifiers. *Biomed. Eng. Online* **2015**, *14*, 54. [CrossRef]
37. Khan, M.H. *Human Activity Analysis in Visual Surveillance and Healthcare*; Logos Verlag Berlin GmbH: Berlin, Germany, 2018; Volume 45.
38. Fan, S.; Jia, Y.; Jia, C. A feature selection and classification method for activity recognition based on an inertial sensing unit. *Information* **2019**, *10*, 290. [CrossRef]
39. Amjad, F.; Khan, M.H.; Nisar, M.A.; Farid, M.S.; Grzegorzek, M. A comparative study of feature selection approaches for human activity recognition using multimodal sensory data. *Sensors* **2021**, *21*, 2368. [CrossRef] [PubMed]
40. Sargano, A.B.; Angelov, P.; Habib, Z. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. *Appl. Sci.* **2017**, *7*, 110. [CrossRef]
41. Hsu, W.C.; Sugiarto, T.; Liao, Y.Y.; Lin, Y.J.; Yang, F.C.; Hueng, D.Y.; Sun, C.T.; Chou, K.N. Can trunk acceleration differentiate stroke patient gait patterns using time-and frequency-domain features? *Appl. Sci.* **2021**, *11*, 1541. [CrossRef]
42. Liang, H.; Sun, X.; Sun, Y.; Gao, Y. Text feature extraction based on deep learning: A review. *EURASIP J. Wirel. Commun. Netw.* **2017**, *2017*, 211. [CrossRef] [PubMed]
43. Fatima, R.; Khan, M.H.; Nisar, M.A.; Doniec, R.; Farid, M.S.; Grzegorzek, M. A Systematic Evaluation of Feature Encoding Techniques for Gait Analysis Using Multimodal Sensory Data. *Sensors* **2023**, *24*, 75. [CrossRef]
44. Khan, M.H.; Farid, M.S.; Grzegorzek, M. A comprehensive study on codebook-based feature fusion for gait recognition. *Inf. Fusion* **2023**, *92*, 216–230. [CrossRef]
45. Lagodzinski, P.; Shirahama, K.; Grzegorzek, M. Codebook-based electrooculography data analysis towards cognitive activity recognition. *Comput. Biol. Med.* **2018**, *95*, 277–287. [CrossRef]

46. Shirahama, K.; Grzegorzek, M. On the generality of codebook approach for sensor-based human activity recognition. *Electronics* **2017**, *6*, 44. [CrossRef]
47. Köping, L.; Shirahama, K.; Grzegorzek, M. A general framework for sensor-based human activity recognition. *Comput. Biol. Med.* **2018**, *95*, 248–260. [CrossRef]
48. Khan, M.H.; Farid, M.S.; Grzegorzek, M. A generic codebook based approach for gait recognition. *Multimed. Tools Appl.* **2019**, *78*, 35689–35712. [CrossRef]
49. Khan, M.H.; Li, F.; Farid, M.S.; Grzegorzek, M. Gait recognition using motion trajectory analysis. In Proceedings of the 10th International Conference on Computer Recognition Systems CORES 2017 10, Polanica Zdroj, Poland, 22–24 May 2017; Springer: Berlin/Heidelberg, Germany, 2018; pp. 73–82.
50. Zhang, J.; Wu, F.; Wei, B.; Zhang, Q.; Huang, H.; Shah, S.W.; Cheng, J. Data augmentation and dense-LSTM for human activity recognition using WiFi signal. *IEEE Internet Things J.* **2020**, *8*, 4628–4641. [CrossRef]
51. Bianchi, V.; Bassoli, M.; Lombardo, G.; Fornacciari, P.; Mordonini, M.; De Munari, I. IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet Things J.* **2019**, *6*, 8553–8562. [CrossRef]
52. Anagnostis, A.; Benos, L.; Tsaopoulos, D.; Tagarakis, A.; Tsolakis, N.; Bochtis, D. Human activity recognition through recurrent neural networks for human–robot interaction in agriculture. *Appl. Sci.* **2021**, *11*, 2188. [CrossRef]
53. Bu, C.; Zhang, L.; Cui, H.; Yang, G.; Wu, H. Dynamic inference via localizing semantic intervals in sensor data for budget-tunable activity recognition. *IEEE Trans. Ind. Inform.* **2023**, *20*, 3801–3813. [CrossRef]
54. Cheng, D.; Zhang, L.; Bu, C.; Wang, X.; Wu, H.; Song, A. ProtoHAR: Prototype guided personalized federated learning for human activity recognition. *IEEE J. Biomed. Health Inform.* **2023**, *27*, 3900–3911. [CrossRef]
55. Huang, W.; Zhang, L.; Wu, H.; Min, F.; Song, A. Channel-Equalization-HAR: A light-weight convolutional neural network for wearable sensor based human activity recognition. *IEEE Trans. Mob. Comput.* **2022**, *22*, 5064–5077. [CrossRef]
56. Saha, J.; Chowdhury, C.; Roy Chowdhury, I.; Biswas, S.; Aslam, N. An ensemble of condition based classifiers for device independent detailed human activity recognition using smartphones. *Information* **2018**, *9*, 94. [CrossRef]
57. Kolkar, R.; Geetha, V. Human activity recognition in smart home using deep learning techniques. In Proceedings of the 2021 13th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 20–21 October 2021; pp. 230–234.
58. Dua, N.; Singh, S.N.; Semwal, V.B. Multi-input CNN-GRU based human activity recognition using wearable sensors. *Computing* **2021**, *103*, 1461–1478. [CrossRef]
59. Khatun, M.A.; Yousuf, M.A.; Ahmed, S.; Uddin, M.Z.; Alyami, S.A.; Al-Ashhab, S.; Akhdar, H.F.; Khan, A.; Azad, A.; Moni, M.A. Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor. *IEEE J. Transl. Eng. Health Med.* **2022**, *10*, 2700316. [CrossRef]
60. Perez-Gamboa, S.; Sun, Q.; Zhang, Y. Improved sensor based human activity recognition via hybrid convolutional and recurrent neural networks. In Proceedings of the 2021 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), Kailua-Kona, HI, USA, 22–25 March 2021; pp. 1–4.
61. Nisar, M.A.; Shirahama, K.; Irshad, M.T.; Huang, X.; Grzegorzek, M. A Hierarchical Multitask Learning Approach for the Recognition of Activities of Daily Living Using Data from Wearable Sensors. *Sensors* **2023**, *23*, 8234. [CrossRef]
62. Pfister, F.M.; Um, T.T.; Pichler, D.C.; Goschenhofer, J.; Abedinpour, K.; Lang, M.; Endo, S.; Ceballos-Baumann, A.O.; Hirche, S.; Bischl, B.; et al. High-resolution motor state detection in Parkinson's disease using convolutional neural networks. *Sci. Rep.* **2020**, *10*, 5860. [CrossRef]
63. Iwana, B.K.; Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE* **2021**, *16*, e0254841. [CrossRef] [PubMed]
64. Rashid, K.M.; Louis, J. Activity identification in modular construction using audio signals and machine learning. *Autom. Constr.* **2020**, *119*, 103361. [CrossRef]
65. Steven Eyobu, O.; Han, D.S. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors* **2018**, *18*, 2892. [CrossRef]
66. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv* **2018**, arXiv:1808.02455.
67. Kalouris, G.; Zacharaki, E.I.; Megalooikonomou, V. Improving CNN-based activity recognition by data augmentation and transfer learning. In Proceedings of the 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; Volume 1, pp. 1387–1394.
68. Wang, J.; Zhu, T.; Gan, J.; Chen, L.L.; Ning, H.; Wan, Y. Sensor data augmentation by resampling in contrastive learning for human activity recognition. *IEEE Sens. J.* **2022**, *22*, 22994–23008. [CrossRef]
69. Tran, L.; Choi, D. Data augmentation for inertial sensor-based gait deep neural network. *IEEE Access* **2020**, *8*, 12364–12378. [CrossRef]
70. Tsinganos, P.; Cornelis, B.; Cornelis, J.; Jansen, B.; Skodras, A. Data augmentation of surface electromyography for hand gesture recognition. *Sensors* **2020**, *20*, 4892. [CrossRef] [PubMed]
71. Jeong, C.Y.; Shin, H.C.; Kim, M. Sensor-data augmentation for human activity recognition with time-warping and data masking. *Multimed. Tools Appl.* **2021**, *80*, 20991–21009. [CrossRef]

72. Cheng, D.; Zhang, L.; Bu, C.; Wu, H.; Song, A. Learning hierarchical time series data augmentation invariances via contrastive supervision for human activity recognition. *Knowl.-Based Syst.* **2023**, *276*, 110789. [CrossRef]
73. Uchitomi, H.; Ming, X.; Zhao, C.; Ogata, T.; Miyake, Y. Classification of mild Parkinson's disease: Data augmentation of time series gait data obtained via inertial measurement units. *Sci. Rep.* **2023**, *13*, 12638. [CrossRef]
74. Guo, P.; Yang, H.; Sano, A. Empirical Study of Mix-based Data Augmentation Methods in Physiological Time Series Data. In Proceedings of the 2023 IEEE 11th International Conference on Healthcare Informatics (ICHI), Houston, TX, USA, 26–29 June 2023; pp. 206–213.
75. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–November 2019; pp. 6023–6032.
76. Cauli, N.; Reforgiato Recupero, D. Survey on videos data augmentation for deep learning models. *Future Internet* **2022**, *14*, 93. [CrossRef]
77. Zhang, B.; Gu, S.; Zhang, B.; Bao, J.; Chen, D.; Wen, F.; Wang, Y.; Guo, B. Styleswin: Transformer-based gan for high-resolution image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11304–11314.
78. Zhao, Z.; Kunar, A.; Birke, R.; Chen, L.Y. Ctab-gan: Effective table data synthesizing. In Proceedings of the Asian Conference on Machine Learning, PMLR 2021, Virtual, 17–19 November 2021; pp. 97–112.
79. Lou, H.; Qi, Z.; Li, J. One-dimensional data augmentation using a Wasserstein generative adversarial network with supervised signal. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1896–1901.
80. Chen, G.; Zhu, Y.; Hong, Z.; Yang, Z. EmotionalGAN: Generating ECG to enhance emotion state classification. In Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science, Wuhan, China, 12–13 July 2019; pp. 309–313.
81. Fons, E.; Dawson, P.; Zeng, X.j.; Keane, J.; Iosifidis, A. Adaptive weighting scheme for automatic time series data augmentation. *arXiv* **2021**, arXiv:2102.08310.
82. Huang, J.; Lin, S.; Wang, N.; Dai, G.; Xie, Y.; Zhou, J. TSE-CNN: A two-stage end-to-end CNN for human activity recognition. *IEEE J. Biomed. Health Inform.* **2019**, *24*, 292–299. [CrossRef] [PubMed]
83. Oh, C.; Han, S.; Jeong, J. Time-series data augmentation based on interpolation. *Procedia Comput. Sci.* **2020**, *175*, 64–71. [CrossRef]
84. Shi, W.; Fang, X.; Yang, G.; Huang, J. Human activity recognition based on multichannel convolutional neural network with data augmentation. *IEEE Access* **2022**, *10*, 76596–76606. [CrossRef]
85. Le Guennec, A.; Malinowski, S.; Tavenard, R. Data augmentation for time series classification using convolutional neural networks. In Proceedings of the ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Grenoble, France, 19–23 September 2022.
86. Wen, T. tsaug. 2019. Available online: https://tsaug.readthedocs.io/en/stable/ (accessed on 20 March 2024).
87. Olah, C. Understanding LSTM Networks. 2015. Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 16 April 2024).
88. Augustinov, G.; Nisar, M.A.; Li, F.; Tabatabaei, A.; Grzegorzek, M.; Sohrabi, K.; Fudickar, S. Transformer-based recognition of activities of daily living from wearable sensor data. In Proceedings of the 7th International Workshop on Sensor-Based Activity Recognition and Artificial Intelligence, Rostock, Germany, 19–20 September 2022; pp. 1–8.
89. Qian, H.; Pan, S.J.; Miao, C. Latent independent excitation for generalisable sensor-based cross-person activity recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 11921–11929.
90. Al-Qaness, M.A.; Helmi, A.M.; Dahou, A.; Elaziz, M.A. The applications of metaheuristics for human activity recognition and fall detection using wearable sensors: A comprehensive analysis. *Biosensors* **2022**, *12*, 821. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.