

Article

# Enhancing Sensor-Based Human Physical Activity Recognition Using Deep Neural Networks

Minyar Sassi Hidri \*, Adel Hidri , Suleiman Ali Alsaif , Muteeb Alahmari  and Eman AlShehri 

Computer Department, Deanship of Preparatory Year and Supporting Studies, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

\* Correspondence: mmsassi@iau.edu.sa

**Abstract:** Human activity recognition (HAR) is the task of classifying sequences of data into defined movements. Taking advantage of deep learning (DL) methods, this research investigates and optimizes neural network architectures to effectively classify physical activities from smartphone accelerometer data. Unlike traditional machine learning (ML) methods employing manually crafted features, our approach employs automated feature learning with three deep learning architectures: Convolutional Neural Networks (CNN), CNN-based autoencoders, and Long Short-Term Memory Recurrent Neural Networks (LSTM RNN). The contribution of this work is primarily in optimizing LSTM RNN to leverage the most out of temporal relationships between sensor data, significantly improving classification accuracy. Experimental outcomes for the WISDM dataset show that the proposed LSTM RNN model achieves 96.1% accuracy, outperforming CNN-based approaches and current ML-based methods. Compared to current works, our optimized frameworks achieve up to 6.4% higher classification performance, which means that they are more appropriate for real-time HAR.

**Keywords:** human physical activity recognition; deep learning; RNN; LSTM; CNN



Academic Editor: Lei Shu

Received: 24 February 2025

Revised: 2 April 2025

Accepted: 6 April 2025

Published: 14 April 2025

**Citation:** Sassi Hidri, M.; Hidri, A.; Alsaif, S.A.; Alahmari, M.; AlShehri, E. Enhancing Sensor-Based Human Physical Activity Recognition Using Deep Neural Networks. *J. Sens. Actuator Netw.* **2025**, *14*, 42. <https://doi.org/10.3390/jsan14020042>

**Copyright:** © 2025 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

Exponential growth has been experienced in recent years regarding the popularity of smartphones, which support a wide variety of functions. Their functionality has rapidly been increased by powerful artificial intelligence (AI) methods and the rise of mobile and ubiquitous computing.

Probably the most interesting and fastest-moving applications of interest to both scientists and industry are activity recognition tasks, including human activity recognition (HAR), which means the computerized method for detecting actions of humans by sensors or cameras. Automation of this process by recent advances in machine learning (ML) [1–7] and deep learning (DL) [8–10] definitely has been used to understand human behavior much better, leading to better decision-making.

Once automated, these systems make use of data from sensors, cameras, and other forms of input to detect and analyze human activities. This technology has become an important tool in various fields like security [11–14], healthcare [15–18], transportation [19–22], etc., due to its potential to improve efficiency and accuracy.

Depending on the data gathering methods and equipment, there are mainly two categories of activity recognition techniques [23–30]: Sensor-based activity recognition and vision-based activity recognition.

Sensor-based activity recognition [16,24,30] makes use of various inertial sensors including magnetometers, gyroscopes, and accelerometers in differentiating various physical

activities. Vision-based activity recognition [11,14,15,29,31] is performed based on the videos or images captured from the cameras. The shortcomings and restrictions of vision-based systems, such as high equipment cost, strong dependence of lighting conditions, and privacy concerns [32–35], have resulted in more attention being paid to the sensor-based technique with better results.

Different applications have been developed in the literature for the recognition and classification of human activity [36]. Most proposed solutions for HAR make use of ML and DL models, since the problems of HAR are considered, in general, to be a classic pattern recognition problem or more precisely, a classification problem [21–24,26–29,32,37].

Most of the related works suffer from reliance on manual feature engineering, ineffective use of temporal data, and suboptimal architecture selection [31,38–41].

The proposed framework covers these gaps by automatic feature learning, effective preprocessing, optimized DL architectures, and extensive validation on the WISDM dataset. What sets it apart is, for the most important one, the temporal exploitation of the sensor data using long short-term memory (LSTM) recurrent neural network (RNNs), beating the benchmarks set by other state-of-the-art models based on convolutional neural networks (CNNs) and traditional models to handle sequential dependencies.

One significant distinction is the architecture's use of LSTM RNNs to take use of the temporal character of sensor data, which performs better than CNNs and conventional models when it comes to managing sequential dependencies. The preprocessing pipeline is also robust, incorporating denoising, normalization, and time window segmentation to improve data quality and model performance.

The paper introduces a key advance over previous work by rectifying certain shortcomings in the traditional approaches and utilizing the strengths of the DL models. These architectures learn features from raw sensor data without manual feature engineering. Besides, the framework allows model architectures to be optimized by appropriate settings of convolutional layers, fine tuning of dropout rates, and activation functions that yield much higher accuracy and computational efficiency.

The main contributions of this article can be highlighted as follows:

- Building three optimal deep neural network-based classifiers, including CNNs, CNN-based autoencoders, and LSTM RNN for HAR.
- Optimizing CNN and LSTM RNN classifiers.
- Validating optimal architectures on the WISDM's (Wireless Sensor Data Mining Lab) <https://www.cis.fordham.edu/wisd़/dataset.php> (accessed on 30 September 2024) and UCI-HAR <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones> (accessed on 1 February 2025) datasets.

The results will then be analyzed to see how each model performs overall, their individual interesting features will be discussed, and it will be seen if the original theoretical belief—that LSTM RNN models work better on time series data compared to CNN and CNN-based autoencoders.

The rest of the paper is structured as follows: Section 2 discusses preliminaries related to HAR. Section 3 highlights the related work in HAR. In Section 4, data preprocessing steps and requirements will be presented. Optimization of deep neural network architectures for HAR is highlighted in Section 5. Section 6 presents an extensive performance evaluation. We present our conclusion and future work perspectives in Section 7.

## 2. Preliminaries

HAR systems depend on the set of activities that must be recognized. This is why grouping activities into categories makes it easier to choose the right mechanisms for identifying them. We distinguish the following three classes of events (see Table 1):

- Short-term events are straightforward undertakings with a set, time-bound length. There are two categories for them. First, body gestures, which refer to outward motions primarily employed as a nonverbal communication tool; and second, transitions, which are the actions that link the performance of two distinct actions, such as sitting followed by surveying.
- Basic activities often exhibit a cyclical or continuous action. Although their lengths/durations vary, they often take far longer than short events. This class yields two categories of activities: dynamic, which are periodic activities like walking and running, and static, which are posture-based activities like sitting and standing.
- Complex activities typically consist of a series of simple activities, brief events that involve multiple subjects or users, or both. For example, two persons must participate in and coordinate numerous moves while performing ballroom dancing.

**Table 1.** Human physical activities typology.

Activity Type	Examples
Gesture (Short events)	Wave hands, Nod, Laugh
Transition (Short events)	Stand up straight to sit
Static (Basic activities)	Standing, Sitting
Dynamic (Basic activities)	Walking, Jogging, Walking upstairs (or Upstairs), Walking downstairs (Downstairs), Running, Cycling
Multiple activities (Complex activities)	Cooking, Body building
Multiple Users (Complex activities)	Talk, Dance, Play

Gyrosopes and accelerometers are two kinds of in-built sensors inside any smartphone used to identify human activity. When employed as a wearable sensor, they give information about the user's angular velocity and linear acceleration, respectively, and are mostly unaffected by outside variables such as the global positioning system's (GPS) inability to receive signals from inside locations or electromagnetic noise in compasses. One of the most widely used sensors for detecting signals of bodily movement is the accelerometer [16]. Its working concept is a seismic mass that moves in response to an acceleration. An electrical signal that can be measured can then be produced from this movement. The development of micro-electro-mechanical systems (MEMS) sensors has made use of this phenomenon. Their technique enables the fabrication of semiconductor-based nanoscale devices. When compared to other sensor technologies, they have the benefit of being low-cost and large-scale to create. The most popular MEMS accelerometers are capacitive sensors based on a cantilever beam with a proof mass whose deflection is related to the acceleration the sensor encounters.

### 3. Related Work

A number of HAR approaches, ranging from traditional ML techniques [42–44] to DL models, have been extensively studied in the literature. Decision trees (DT) [45,46], random forests (RF) [45,46], k-nearest neighbors (k-NN) [47,48], artificial neural networks (ANN) [49], support vector machines (SVM) [50–52], and probabilistic approaches like Naïve Bayes (NB) [53] and Hidden Markov Models (HMM) [54,55] have been utilized in the majority of the approaches. They were heavily optimized on hand-designed features, which are difficult to design and introduce bias from outside, though accurate enough to utilize.

DL introduction has eschewed such constraints using the capability to extract features automatically. Comparative studies of ML approaches versus DL approaches have been able to establish the effectiveness of DL models when there is copious data available [56].

For instance, CNNs and RNNs, particularly LSTM networks, have been demonstrated to be more effective than traditional ML models at capturing spatial and temporal correlations in sensor data [56–58].

Hybrid DL architectures were also demonstrated in recent studies. CNN-LSTM architectures apply CNNs for spatial feature extraction and employ LSTM layers to recognize sequential patterns and attain significantly improved accuracy [59].

Recent works have underlined that state-of-the-art architectures, such as transformers and attention-based models, outperform traditional ones by a large margin. They have been especially effective, with one transformer model achieving an impressive 99.2% for accuracy on smartphone motion sensor data, as opposed to the conventional methodology that could reach only 89.67% [60]. Feature fusion techniques, such as integrating outputs from Graph Convolutional Networks (GCN) and transformers, have further enhanced the accuracy significantly, with quite significant improvements noted for datasets like HuGaDB and TUG [61]. Comparative performance analyses further reveal the dominance of emerging architectures, as a recent study demonstrated a novel model achieving 84.09% accuracy, outperforming established approaches like CNNs and LSTMs, signaling a shift towards more robust models [62].

Table 2 presents comparative analysis of HAR approaches considering different methodologies, strengths, and limitations.

**Table 2.** Comparative analysis of HAR approaches.

Approach	Feature Extraction	Model Type	Strengths	Weaknesses
DT [45,46]	Manual	ML	Simple, interpretable, and low computational cost.	Prone to overfitting and less effective with large datasets.
RF [45,46]	Manual	ML	Reduces overfitting and handles large feature sets.	Computationally expensive and requires parameter tuning.
k-NN [47,48]	Manual	ML	Simple and no training phase required.	High memory usage and slow for large datasets.
SVM [50–52]	Manual	ML	Effective in high-dimensional spaces.	Computationally intensive for large datasets.
NB [53]	Manual	Probabilistic	Fast and works well with small datasets,	Assumes feature independence and less accurate.
HMM [54,55]	Manual	Probabilistic	Captures temporal dependencies.	Requires large datasets and sensitive to noise.
ANN [49]	Manual	DL	Can model complex patterns.	Requires large labeled datasets and prone to overfitting.

**Table 2.** Cont.

Approach	Feature Extraction	Model Type	Strengths	Weaknesses
CNN [56,63]	Automatic	DL	Excels at spatial feature extraction.	Struggles with long-term dependencies.
LSTM [58]	Automatic	DL	Captures long-term temporal dependencies.	High computational cost and requires more training data.
CNN-LSTM [59]	Automatic	DL	Combines spatial and temporal feature learning.	Computationally expensive and requires large memory.
Transformers [60]	Automatic	DL	Handles long-term dependencies effectively.	High complexity and requires large-scale training data.
GCN [61]	Automatic	DL	Captures structural relationships in data,	Not widely used for HAR and computationally expensive.

Despite these advances, there remain issues in the optimization of DL architectures for HAR with respect to handling imbalanced datasets, sensor variability, and real-time demands [64–66]. Our study surpasses these shortcomings by proposing and optimizing CNN, CNN-based autoencoder, and LSTM RNN architectures with improved feature learning and classification accuracy.

The key innovations and contributions of our study are:

- Optimized DL architectures: Unlike existing approaches, we will employ a combination of CNN, CNN-based Autoencoder, and LSTM RNN models specifically optimized for HAR.
- Temporal exploitation: We will leverage LSTM RNNs to better capture the sequential dependencies of sensor data, improving classification accuracy over CNN-only models.
- Robust pre-processing: The preprocessing steps to be performed include denoising, normalizing, and time window segmentation to improve data quality, as well as the strength of the model.

#### 4. Data Preprocessing

The primary focus of our work is to optimize DL models specifically for the WISDM dataset. The data was provided by the WSDM Lab. It is issued from 36 smartphone users at a sampling rate of 20 samples per second. As the user performs six different activities during the experiment, the data contains acceleration values on the *x*, *y*, and *z* axes.

1. X-axis: left to right.
2. Y-axis: top to bottom.
3. Z-axis: front to back.

1,098,207 physical activities are presented with the used dataset. The sampling rate for its measured data is 20 Hz and the range within which the accelerometer measured the values is  $\pm 2$  g. This dataset classified the physical activities into 6 different classes, as presented in Table 3.

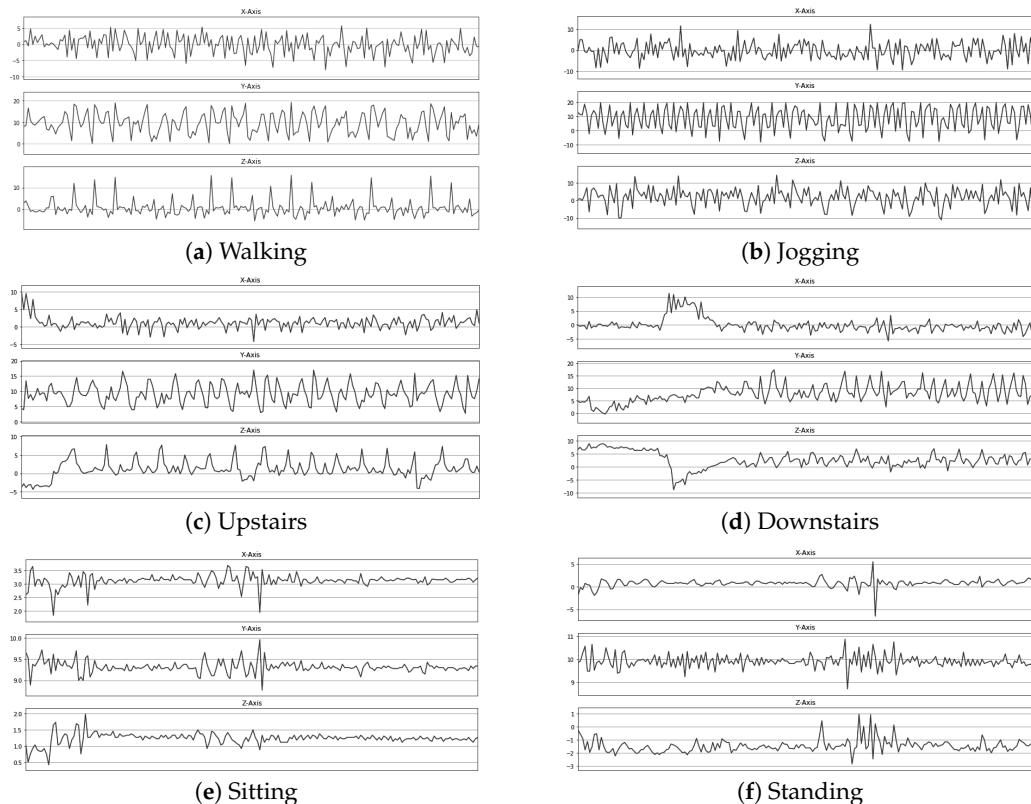
**Table 3.** The distribution of activities within the WISDM dataset.

Activity	Number of Sample	Proportion
Walking	424,400	38.6%
Jogging	342,177	31.2%
Upstairs	122,177	11.2%
Downstairs	100,427	9.1%
Sitting	59,939	5.5%
Standing	48,395	4.4%

According to Table 3, there is an inconsistency between the total number of activity samples reported in the dataset, which is 1,098,207, and the sum of activities, which is 1,057,515. This difference in 40,692 samples from the total number of samples for activities to the sum of activities from Table 3 is because of the following data pre-processing steps, which are very vital in ensuring the quality of the data and model reliability:

- Cleaning of noisy data: we removed the samples that were flagged as noisy or corrupted during the initial data analysis phase. These samples had irregularities such as extreme outliers, missing values, or implausible accelerometer readings that could affect model training and evaluation.
- Exclusion of unlabeled samples: a portion of this dataset contained samples with missing or invalid labels. Such entries were excluded to maintain the integrity of the classification task since the absence of correct labels would certainly affect model performance.
- Consistency filtering: The dataset was filtered against consistency criteria in order to maintain only those samples that were perceived to correspond to the physical activities used for labeling.

Figure 1 shows examples of what the triaxial accelerometer values look like for each physical activity.

**Figure 1.** Examples of what the triaxial accelerometer values look like for each physical activity.

It is also necessary to separate the data into features and labels and then into training and testing sets. In order to feed labels into the classifier, they must be encoded once.

## 5. Optimization of Deep Neural Networks Architectures for HAR

The focus of this work is to create DL models that can accurately predict human physical activity through accelerometer data. This will be tested with three different classifiers: a CNN classifier, a CNN-based autoencoder classifier, and an LSTM RNN classifier. The steps for this are threefold:

- Creating the three optimal classifiers to recognize physical human activities.
- Validating the models by testing them on a labeled dataset.

The results will then be analyzed to see how each model performs overall, their individual interesting features will be discussed, and it will be seen if the original theoretical belief—that LSTM RNN models work better on time series data compared to CNN and CNN-based autoencoders.

### 5.1. Cnn-Based Optimal Classifier

CNNs have proven to be highly effective in various computer vision tasks, including image recognition, object detection, image segmentation, and more.

The key components and characteristics of a CNN are:

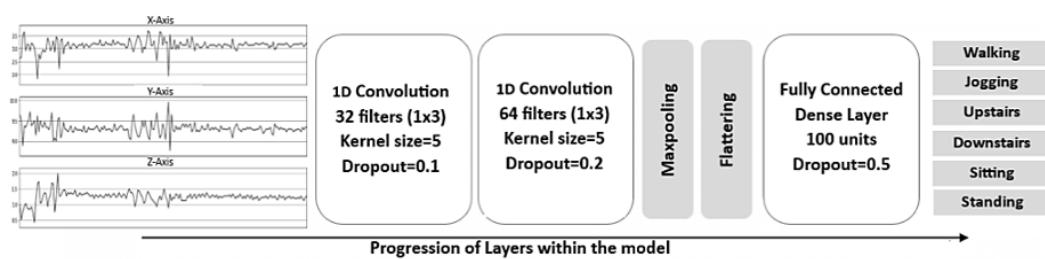
- Convolutional layers: These layers apply convolution operations to the input data. Convolution involves sliding a filter (also known as a kernel) over the input to detect spatial patterns and features. This allows the network to automatically learn hierarchical representations.
- Pooling layers: Pooling layers downsample the spatial dimensions of the input by reducing the resolution. Common pooling operations include max pooling, where the maximum value in a local region is retained, and average pooling, which takes the average.
- Activation functions: Non-linear activation functions, such as Rectified Linear Unit (ReLU), are applied to introduce non-linearity into the network. ReLU is commonly used in CNNs due to its simplicity and effectiveness.
- Fully connected layers: These layers connect every neuron from one layer to every neuron in the next layer. Fully connected layers are typically found at the end of the network and are responsible for producing the final output.
- Training with backpropagation: CNNs are trained using backpropagation and gradient descent. During training, the network adjusts its weights and biases to minimize the difference between predicted outputs and actual targets.
- Local Receptive Fields: CNNs exploit the local relationships between nearby pixels by using small squares of input data in the convolutional layers. This allows them to capture spatial hierarchies and patterns.
- Weight sharing: Parameters (weights and biases) are shared across different parts of the input, which reduces the total number of parameters and makes CNNs computationally efficient.

Figure 2 shows the proposed CNN architecture used to recognize the different human activities.

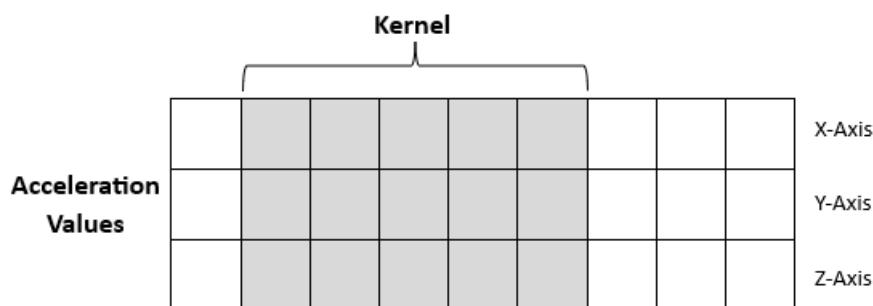
The parameters of each layer from Figure 2 are described below:

- *Input:* We must prepare and give the acceleration data to the CNN as a 2-dimensional input as it is captured as time series data from the smart device that was used to measure it. Time is the first dimension, while the acceleration's value is the second. It is demonstrated how to do this in Figure 3.

- *Convolution Layers:* The two convolutional layers get the acceleration data after they have been preprocessed into a 2D structure. The ReLU activation function is used in the first convolutional layer, which contains 32 convolutional filters and a kernel size of 5. Similarly, the 2nd convolutional layer also has a kernel size of 5, also uses the ReLU activation function, however, has a convolutional filter of 64 convolutional filters, which refers to how many pixels the kernel shifts by, is 1. The 1st convolution layer uses a dropout rate of 0.1, while the second convolution layer has a dropout rate of 0.2.
- *Maxpooling layer:* The pooling layer has a size of 2.
- *Fully connected dense layer:* The Maxpooling layer's output is intended to be flattened. After that, it is passed into a layer of 100 neurons that are highly linked. To prevent over-fitting, this layer has a dropout rate of 0.5. With just 5 neurons employed to represent the 6 output activities the neural network is meant to identify, another thick layer representing the output layer is added. Currently, the likelihood of the output activity that the data at the current iteration most likely reflects is calculated using a softmax activation function.



**Figure 2.** Optimal CNN model architecture.



**Figure 3.** Kernel sliding over input data.

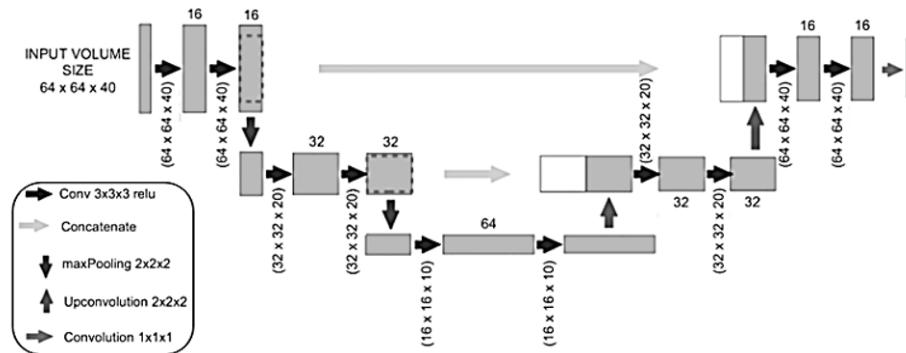
### 5.2. Cnn-Based Autoencoder Optimal Classifier

A CNN-based autoencoder combines the architecture of a CNN with the principles of an autoencoder. An autoencoder is a type of neural network designed for unsupervised learning and dimensionality reduction. The architecture consists of an encoder that compresses the input data into a lower-dimensional representation, and a decoder that reconstructs the original input from this representation. In the case of a CNN-based autoencoder, convolutional layers are used for feature extraction.

This combination makes CNN-based autoencoders well-suited for tasks where learning hierarchical features and spatial relationships in the data are essential.

The network design shown in Figure 4 was adopted from [67]. It was composed of a decoding channel (right side of Figure 4) and an encoding channel (left side). With two consecutive convolutions of  $3 \times 3 \times 3$  (with a padding of 2) applied repeatedly, the encoding path followed the standard CNN architecture. Each convolution was followed by an activation per rectified linear unit (ReLU) [68] and a *maxPooling* operation of  $2 \times 2 \times 2$ .

Starting with 16 filters at the initial convolution, the number of filters was doubled at each subsampling (encoding) step. Every stage of the expansion route involved  $3 \times 3 \times 3$  convolution after the feature map had been upsampled.



**Figure 4.** CNN-based autoencoder optimal architecture.

The last layer included mapping each activation map to the correct number of classes (in this example, the number of activities, represented by a convolution of  $1 \times 1 \times 1$ ). There were 11 convolution layers in the network overall, and 387,889 parameters required to be calculated. Following the U-net-predicted mask was a morphological dilation with  $3 \times 3 \times 3$  square connectivity. In order to align with the process carried out for the ground truth, we also employed fixed thresholding, where the threshold was set at 1.3 times the average value of a hemispherical swap of the projected U-net mask.

### 5.3. Lstm RNN-Based Optimal Classifier

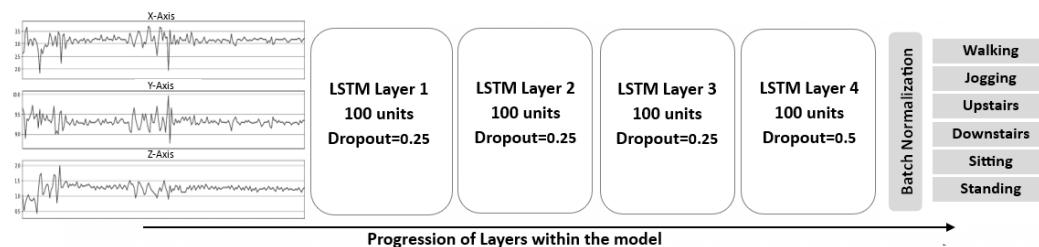
An LSTM RNN is designed for handling sequential data and capturing long-term dependencies. An optimal classifier using LSTM can be applied in our task because the input data has a temporal or sequential structure.

These networks have memory cells and gating mechanisms that allow them to store and selectively update information over time. They are well-suited for processing and classifying sequences of data.

A stacked LSTM refers to the use of multiple LSTM layers stacked on top of each other in a neural network architecture. This is a form of DL where the output of one LSTM layer serves as the input to the next layer, allowing the model to learn hierarchical and more complex representations of sequential data.

Stacking LSTM layers enables our model to capture hierarchical features and dependencies in sequential data. Each layer can learn different levels of abstraction.

The stacked architecture used for the LSTM RNN for the multiple classification of HAR is shown in Figure 5.



**Figure 5.** Stacked architecture of used LSTM RNN model: Optimized model.

The parameters used in the above architecture are explained below:

- *Input:* Time series data are the accelerometer readings used to train the model. It is important for the LSTM within the model that this data is prepared appropriately for it. The 3 axis accelerometer data needs to be reshaped into a parallel 3D structure—as is needed to efficiently build and train the LSTM network. 200 samples were fed into the network to form a segment or a group. The value chosen as the label for a group of sampled data is based on the mode—the most commonly occurring label—within said group. The previous 199 samples would act as memory for the LSTM network.
- *LSTM Layer:* Four LSTM network layers were added to the network. Each layer had 100 units or neurons added to them. Dropout of 25% was added to each individual LSTM layer to prevent over fitting.
- *Batch Normalization:* Batch normalization helps in speeding up the training process hence batch normalization is added to the model.
- *Fully connected layer:* A dense layer is added to complete the model. 5 units are added to this layer to represent the 6 different classifications.

DL model optimization wasn't an easy extension but involved massive amounts of trial and error as well as experimentation. Some of the key components of the used iterative process were:

- Hyperparameter tuning: Selecting the optimal learning rates, batch sizes, dropout rates, and activation functions involved multiple rounds of experimentation. Early settings used to lead to overfitting or slow convergence, and some tweaks were required for improved generalization of models.
- Representation of features: The preprocessing pipeline, from normalization to time-window segmentation, was optimized for extracting valid features. Earlier attempts with smaller window sizes showed degraded performance, so balancing temporal resolution and computational cost.
- Model architecture changes: Initial CNN and LSTM models suffered from issues such as vanishing gradients (in deep LSTMs) and inadequate spatial feature learning (in CNNs). This is why modifications such as batch normalization, residual connections, and increased filter sizes were introduced to stabilize the model and improve its performance.
- Computational constraints: The trade-off between model complexity and computational feasibility was a key consideration. More accurate results were provided by deeper models, but at the cost of inference time. Techniques such as model pruning and weight sharing were explored to minimize efficiency loss without compromising accuracy.
- Cross-dataset generalization: While the models performed well on WISDM, preliminary experiments on other datasets (e.g., UCI-HAR) indicated domain adaptation problems due to variations in sensor characteristics and activity distributions. This highlighted the need for future research directions in transfer learning and domain adaptation.

## 6. Computational Results

### 6.1. Experimental Setup

The computational environment consisted of a Single Nvidia Tesla K80 GPU, 12 GB of RAM, and a 1 TB NVMe SSD. The software environment included Python 3.13.0 as the programming language, and TensorFlow 2.6.0 with the Keras API as the DL framework [69].

The batch size was set to 128, the learning rate was initialized to 0.001 using the Adam optimizer, with the default values for the other parameters:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1 \times 10^{-7}$ . The loss function employed was categorical cross-entropy, and the models were trained for 100 epochs. To ensure optimal performance and reduce the risk of

overfitting, we applied several techniques: early stopping was implemented to monitor the validation loss with a patience of 10 epochs, dropout regularization with rates ranging from 0.2 to 0.5 was employed across different layers, and Xavier initialization was used to set the initial weights.

For CNN, a sequential model was used for the implementation because it allows the creation of a model and then adding layers to it later on. That greatly simplifies the creation of DL models. 1D convolution layers, 1D MaxPooling layer, flattening layer, dropout, and dense layers were added to the sequential model. The parameters of each of these layers have been explained previously.

The LSTM RNN was a sequential model comprising several layers of LSTM, dense, dropout, and batch normalization. The input layer consisted of 100 neurons. Then, 3 more LSTM layers were added on top of each other, allowing the pattern and dependencies between the input data and their corresponding activities to be built up. Lastly, there was an output layer at the end, consisting of 6 neurons for the 6 different classifications. The final model was trained with a mean squared error loss, with the default learning rate of 0.002, using the Adam optimizer for 100 epochs.

The network input, such as temporal dependencies, has been fed using a sliding time window that extracts separate data segments. To gain better accuracy, the window width and step size have been modified and optimized. Because an activity label is assigned to every time step, for each segment, the label selected would be the one with higher frequency. As a rule, it's window width or time segment, which here is 200 while the time step is 100.

The data is split into two subgroups: *rain data* and *test data*, in the ratio of 80:20. Similarly, the training data is again divided into training and validation data. Then, the generated HAR model is trained using the training data and validated on the validation data. We tested the pretrained model using 20% random samples of the dataset in order to assess the performance of the network.

## 6.2. Results

Figure 6 shows the confusion matrix for a CNN model's performance on the validation set for classifying human activities. The matrix visualizes the model's accuracy in predicting different the different activities (*Downstairs*, *Jogging*, *Sitting*, *Standing*, *Upstairs*, *Walking*). On validation set, the CNN model achieves an accuracy of up to 92.1%.

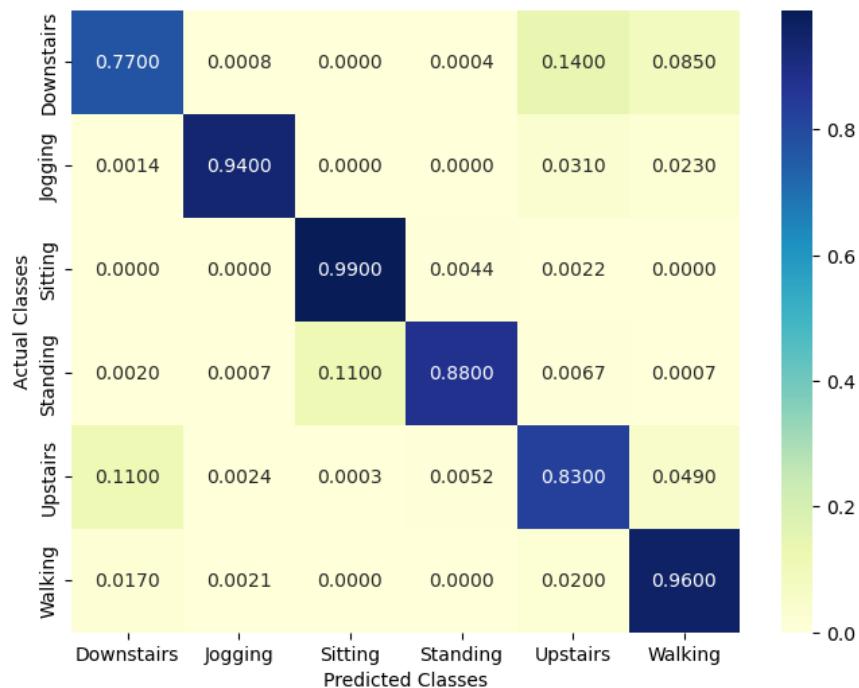
As shown in Figure 6, the model shows high accuracy in identifying *Sitting* (0.99), *Jogging* (0.94), and *Walking* (0.96). The model struggles more with *Upstairs* with 0.83 and *Downstairs* with 0.77. Even if the score is relatively good in *Standing*, at 0.88, it misclassifies a notable number of instances with other activities-mostly with *Sitting*. Indeed, though the model is able to find a quite good part of the *Downstairs* examples, with a remarkable portion misclassifying into walking and then, at a much lesser rate, to other activities.

Interesting to notice in the results is that stationary activities such as *Sitting* have much higher accuracy compared to mobile activities like *Walking* etc. This is possibly because a person, while *Siting*, does less movement of the wrists-where the Accelerometer was placed, whereas when said person would be moving, his wrist position can change quite a lot, which can make the accelerometer values to change drastically thus not allowing to recognize a pattern.

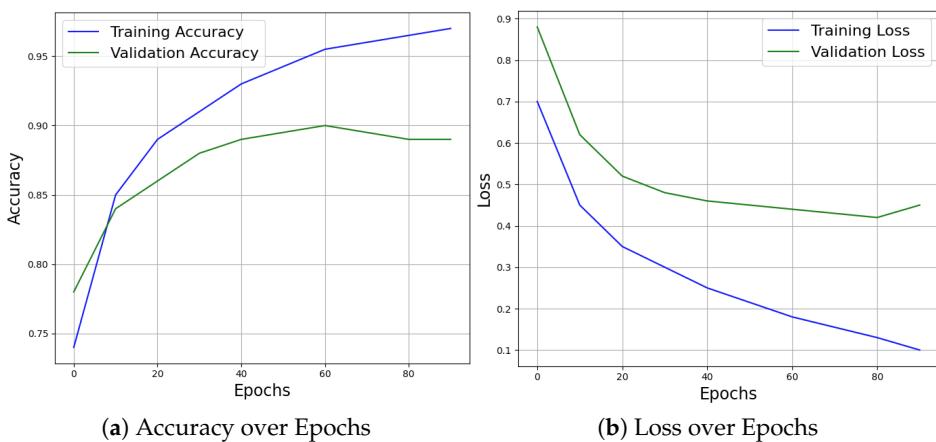
Either *Upstairs* or *Downstairs* are predicted as *Walking* or *Standing*-both of which are positions where the wrist positions of people might be similar to not only each other but to *Upstairs* or *Downstairs* as well. The positive result over here is that it doesn't classify either of *Upstairs* or *Downstairs* as *Sitting*—a position where the posture and wrist position

must be very different compared to the rest. Therefore, the classifier can find a pattern in activities which involve stationariness but fail to do the same for mobile activities.

Figure 7a shows how well the model learned on the test data as the number of epochs were increased. The model shows that it learns well over time as the accuracy on the training set increases from 80% to 96% over 90 epochs. Moreover, the accuracy on the testing set has been improved in this range of epochs. The model seems to be consistent in its accuracy on the testing set as it remains approximately at 91%.



**Figure 6.** Confusion matrix of CNN model's predictions on the validation set.



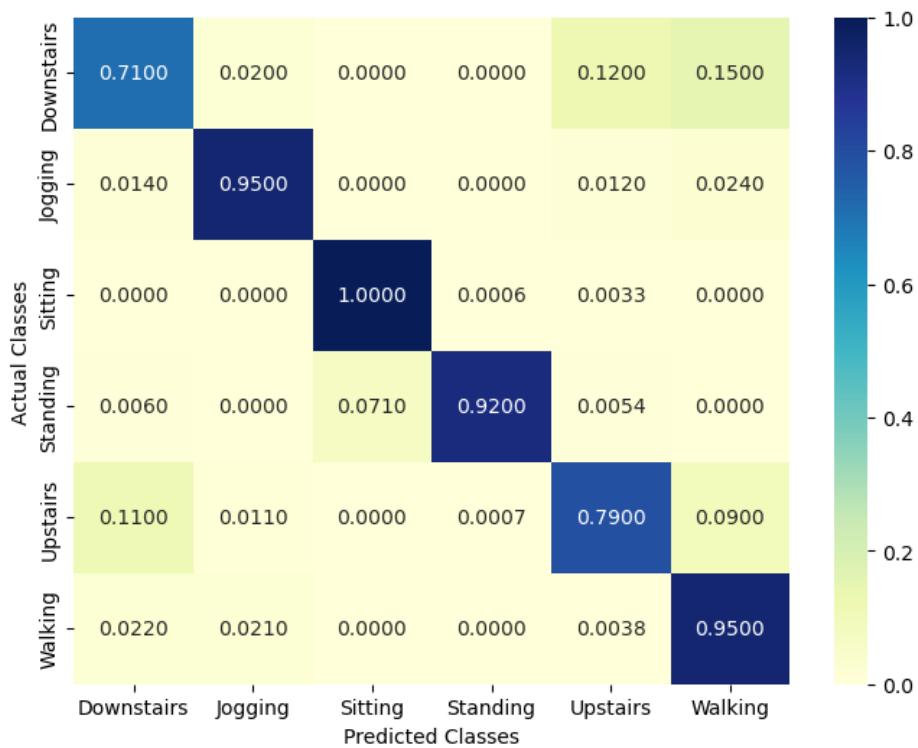
**Figure 7.** CNN: Training & validation.

The training and testing sets' losses decrease with time, as seen in Figure 7b. A model is better the smaller its loss (unless it has overfitted to the training set). The model's performance for these two sets is shown by the loss, which is computed based on training and validation. Loss is not a percentage, in contrast to accuracy. It is an accumulation of all the mistakes produced in training or validation sets for every example.

Our primary goal in the learning model is to change the weight vector values using various optimization techniques in order to lower (minimize) the loss function's value with regard to the model's parameters. The derived Loss values indicate how well or poorly

a certain model performs following each optimization cycle. The decrease of loss should ideally occur after one or more iterations.

The CNN-based autoencoder model achieves an accuracy of up to 90.5% on the testing dataset. The confusion matrix of CNN-based autoencoder classifier is given in Figure 8.



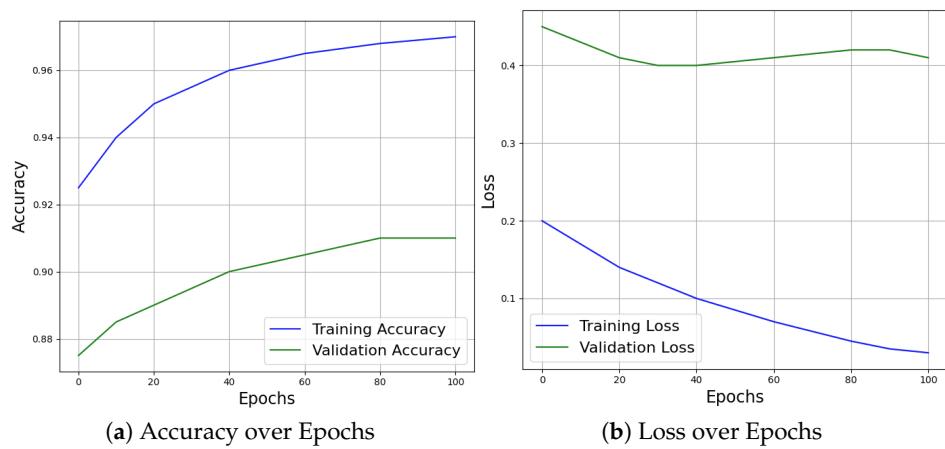
**Figure 8.** Confusion matrix of CNN-based autoencoder model's predictions on the validation data.

Similar to the standard CNN, results shown in Figure 8 are also really strong at *Sitting* (1.0), *Jogging* (0.96), and *Walking* (0.95). For this model, compared to Figure 6, there is slightly lower accuracy for *Downstairs* (0.71) and *Upstairs* (0.79). The confusion on *Standing* is also higher in this model than in the standard CNN. A greater amount of *Standing* is classified incorrectly, particularly as *Sitting*.

Both models show generally good performance and are very good in classifying *Sitting*, *Jogging*, and *Walking*. Both struggle with *Downstairs* and *Standing*. Though for the autoencoder-based model, similar strong performance for a number of classes is present, there is slightly decreased performance, as indicated by a larger amount of misclassifications for *Standing* and *Upstairs*.

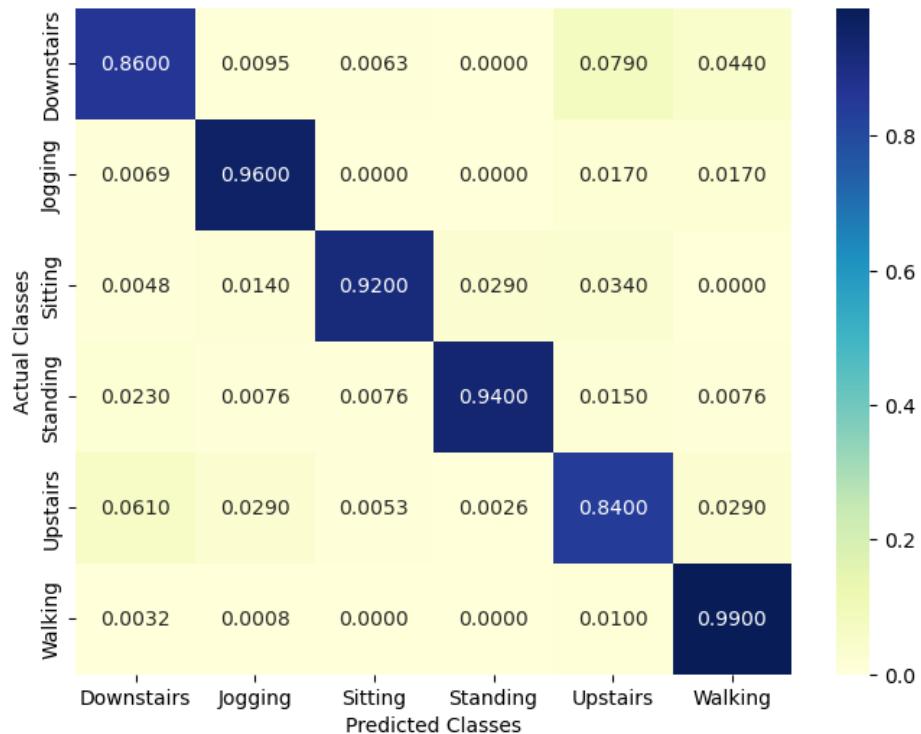
Both models classify the data from this HAR task really well, while neither provides superior classification with regards to classes *Standing* and *Downstairs*. This type of matrix makes more visible kinds of mistakes that each model performs; these further give reason to model modification with feature engineering or architecture modification.

Figure 9a shows how well the model learned on the test data as the number of epochs were increased. The model shows that it learns well over time as the accuracy on the training set increases from 92.4% to 97% over 100 epochs. Moreover, the accuracy on the testing set has been improved in this range of epochs (from 86.5% to 91%). The model seems to be consistent in its accuracy on the testing set as it remains approximately at 90.5%.



**Figure 9.** CNN-based autoencoder: Training & validation.

The accuracy of the LSTM RNN classifier is 96.1%; however, it might potentially be somewhat enhanced by reducing the sliding window step size. The confusion matrix from Figure 10 is used to analyze the results for each particular activity.



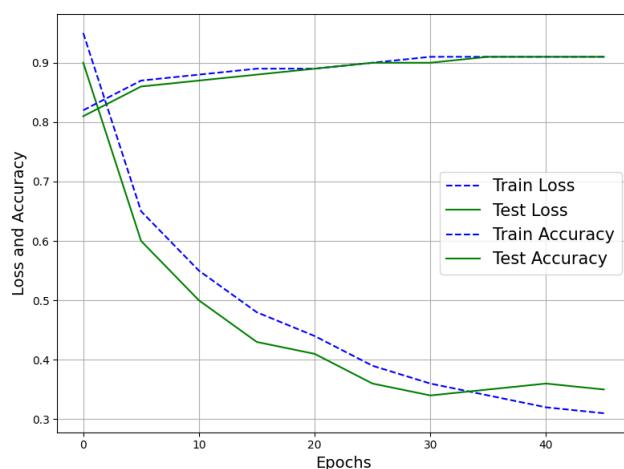
**Figure 10.** Confusion matrix of LSTM RNN model's predictions on the validation data.

Similarly, the same trend is continued in the case of the LSTM RNN model. The various activities such as *Jogging*, *Sitting*, *Standing*, and *Walking* can be classified quite well, whereas *Downstairs* and *Upstairs* activities are not well classified.

Contrary to the performance of the CNN on the validation dataset, the model did not succeed in identifying upstairs and downstairs activities more correctly. The LSTM RNN model performs better on the validation dataset. This verifies the theoretical assumption made before these models were created that LSTM RNNs, conceptually, do a better job in classifying time series data compared to CNNs. For *Walking*, LSTM and CNN models perform at 99% and 96% respectively. But for *Sitting* and *Standing*, the CNN model performs much better compared to the LSTM RNN.

Figure 11 shows the changes in model accuracy on both training and test sets over a running range of epochs. The highest accuracy for the model in the training set comes at 40–50 epochs where there is less disparity between the testing set accuracy and training set accuracy. Training and validation accuracy increased, with the training accuracy a bit higher than that of the validation, but not to an extreme limit, meaning fairly reasonable overfitting. It means that whereas the model is learning well in training, it does not fully generalize to the data that it has not seen. The accuracy of training and validation reaches a plateau at approximately 0.95 and 0.92, respectively.

The loss plot shows a decrease in both training and validation loss during training. Again, the training loss is lower than the validation loss, which is typical, but the difference is reasonably small, further supporting the conclusion of moderate overfitting rather than significant overfitting. The preprocessing steps described previously contributed to improved model performance by ensuring that only high-quality, labeled samples were used.



**Figure 11.** Accuracy and the loss function of the LSTM RNN model against the number of epochs.

While the primary focus of this study was to optimize DL models specifically for the WISDM dataset, we acknowledge the importance of evaluating generalizability across diverse datasets. The WISDM dataset offers a comprehensive set of accelerometer-based readings from 36 users performing six distinct activities, making it a benchmark dataset widely used in HAR research. This allows for direct comparison with prior studies and provides a robust baseline for model evaluation. The data preprocessing steps, as well as model optimizations (CNN, CNN-based autoencoder, and LSTM RNN), were tailored to maximize performance on this dataset while ensuring consistency and reproducibility.

In order to understand better where the most disparate models are, we calculated in Table 4 standardized residuals to highlight where the greatest deviations are. Standardized residuals measure how far each observation is from the expected count in the Chi-square test.

**Table 4.** Standardized residuals analysis for model performance.

Model	Correct Classifications Residual	Incorrect Classifications Residual
CNN	+0.46 (slight above expected)	-1.15 (slight below expected)
CNN-based autoencoder	-3.71 (significantly lower than expected)	+9.24 (significantly higher than expected)
LSTM RNN	+3.25 (significantly higher than expected)	-8.08 (significantly lower than expected)

According to Table 4, LSTM RNN best correct classification (better than predicted) and minimum misclassification rate (much improved over predicted) have the best values. It confirms it's better than others. CNN-based autoencoder worse than predicted with fewer correct classification and significantly more errors than predicted. CNN fairly neutral; never deviates very far from prediction.

In summary, LSTM RNN performs much better than CNN and CNN-based autoencoder, confirming it is the top model. CNN-based autoencoder is the worst, with a much worse-than-expected error rate. CNN is in between, performing as expected.

We present in Table 5 the performance comparison results with WISDM dataset. The obtained accuracy is higher than that of previous research using the suggested optimizations offered in the baseline architectures (LSTM RNN).

**Table 5.** Performance comparison results with WISDM dataset.

Model	Accuracy
Xu et al. [56]	91.97%
Pienaar and Malekia [57]	90%
Yu et al. [58]	94%
Sheng et al. [63]	95.52%
Prasad et al. [70]	89.67%
<b>Optimal LSTM RNN</b>	<b>96.1%</b>

To give a more holistic assessment of the generalizability and robustness of the proposed architecture, we validate the proposed model on additional benchmark HAR datasets such as the UCI HAR dataset (<https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>; accessed on 1 February 2025), which encompasses different sensor setups, activity types, and environmental conditions.

The types of sensors used in the two datasets significantly affect the nature of the data and further model performance. In UCI-HAR, the embedded inertial sensors (accelerometer and gyroscope) from the smartphone themselves are mainly tri-axial for activity recognition. WSDN considers wireless sensing technologies like WiFi and ultra-wideband signals. This represents a wide range of sensors, from the very precise tracking of motion provided by inertial sensors to the wider environmental contexts possible with wireless sensing. There is also considerable variation in the diverse environments and activities that these datasets capture.

Although it may include some unpredictability and noise, WSDN considers less controlled surroundings and hence captures intricate and varied activities that could better mimic real-world circumstances. Despite having the same HAR goal, the difference in approach and sensors between the two datasets leads to different classification accuracy and practical usefulness. UCI-HAR represents data from structured environments with well-defined activities such as *Walking*, *Walking Upstairs*, *Walking Downstairs*, *Sitting*, *Standing*, and *Laying*. The dataset is collected from 30 people, performing different activities with a smartphone to their waists. The data is recorded with the help of sensors (accelerometer and gyroscope) in that smartphone.

Table 6 presents a performance comparison between the 4-layer CNN-LSTM model and the optimal LSTM-RNN model for human activity recognition (HAR) using the UCI-HAR dataset. The overall accuracy reached is 99.6%.

Both models achieve perfect classification (accuracy = 1) for *Walking*, *Walking Upstairs*, *Walking Downstairs*, *Standing*, and *Laying* activities. For the *Sitting* activity, the 4-layer CNN-LSTM achieves an accuracy of 0.983, slightly lower than the 0.987 achieved by the

optimal LSTM-RNN. This indicates that the optimal LSTM-RNN has a marginal advantage in distinguishing the *Sitting* activity.

**Table 6.** Performance comparison results with the UCI-HAR dataset.

Activity	4-Layer CNN-LSTM [59]	Optimal LSTM-RNN
Walking	1	0.989
Walking Upstairs	1	1
Walking Downstairs	1	1
Sitting	0.983	0.987
Standing	1	1
Laying	1	1
Accuracy	99.71	99.6%

The 4-layer CNN-LSTM model achieves an overall accuracy of 99.71%, while the optimal LSTM-RNN model achieves a slightly lower accuracy of 99.6%. This reflects that both models are highly effective for human activity detection using the UCI-HAR dataset, with negligible differences in their overall performance.

While the primary focus of this article was to optimize DL models specifically for the WISDM dataset, we also demonstrate the efficiency of the proposed architecture in detecting activities using UCI-HAR dataset.

### 6.3. Discussion

Activities such as *Walking*, *Jogging*, and *Sitting* have a very high accuracy, whereas the significant number of misclassifications among the three classes *Standing*, *Upstairs*, and *Downstairs*. This is not very surprising, because there is an inherent similarity in the sensor information collected for these three activities.

In all models, the confusion matrices show a considerable number of instances where *Standing* is misclassified as *Sitting* and vice versa. This is probably due to similar postural characteristics. Both activities involve relatively static body positions, hence changing the acceleration minimally, which the model fails to highlight. Sensor limitations with respect to subtle movements, besides the difficulty in distinguishing between minimal changes in acceleration, may also present some challenges.

The errors involving the *Upstairs* and *Downstairs* classifications arise from the similarities in the rhythmic nature of the movement and their similar frequency spectrum. Both involve changes in posture and motion but may have overlapping frequency ranges that the model has trouble separating without additional, perhaps more subtle, features.

While our models are good at recognizing structured human activities such as *Walking*, *Jogging*, and *Sitting*, they may require further adaptation or supplementation to accurately classify short-term gestures. We also suggested some potential future research directions, such as multi-modal sensor fusion or finer time segmentation, to improve classification for transient activities. This limitation is particularly relevant for gestures, which will often require higher sampling rates, additional sensor modalities (e.g., rotation using a gyroscope), or domain-specific models such as attention-based models in an effort to improve recognition performance.

To address this, we revised the manuscript by clarifying that while our models excel in recognizing structured human activities such as walking, jogging, and sitting, they may require further refinement or complementary methods for accurately classifying short-term gestures. Additionally, we suggested future research directions, such as multi-modal sensor fusion or finer-grained time segmentation, to improve classification for transient activities.

Although our suggested DL architectures—CNN, CNN-based Autoencoder, and LSTM RNN—performed best in HAR using the WISDM dataset, the following limitations should be taken into account:

- Dataset-specific performance: Models were trained and tested using the WISDM dataset, which can be devoid of variability present in actual HAR situations. Sensor placement variability, user group variability, and environment variability between datasets can affect the model's generalizability.
- Limited sensor modalities: Accelerometer data only is utilized in the paper. Though adequate for HAR, incorporation of other sensors such as gyroscopes and magnetometers would provide additional feature representations, hence improved classification performance for movements with fewer differences in motions.
- Sensor placement variability: The data is assumed to have a consistent smartphone placement (often in the pocket). In reality, use is less consistent, phones are held in different positions (e.g., hand, wrist, backpack), affecting data patterns. Domain adaptation methods for model robustness for placements need to be researched in the future.
- Class imbalance: WISDM dataset is class imbalanced with standing and sitting under-represented. Preprocessing is already performed, but there are other methods such as synthetic data augmentation or weighted loss that may be employed in order to enhance performance further.
- Computational complexity: Deep models, particularly LSTM RNNs, are extremely computationally intensive. Optimization algorithms such as model pruning, quantization, or knowledge distillation can possibly be utilized while deploying such models on edge devices or low-processing-capacity phones.
- Real-time constraints: Although accuracy was our priority, real-time inference speed and latency were not investigated to a large degree. Investigating the trade-offs between model complexity and deployment efficiency is an important area of future work.

Future work will be engaged in overcoming these limitations by performing cross-dataset validation, multimodal sensor fusion, adaptive learning methods, and computationally efficient DL models for real-time deployment.

## 7. Conclusions

Deep neural networks such as CNNs and RNNs have lately shown their powers by automatically learning characteristics from the raw sensor data, even achieving state-of-the-art results.

This work focuses on optimizing DL models that can accurately identify the physical activities of human beings based on the accelerometer data collected from smartphones. It is achieved by creating an LSTM RNN and a CNN model that will be trained on the WSDM's dataset.

To validate the ability of the model to recognize the physical activity of individuals through accelerometer data, the different models were been tested on labeled datasets provided by WISDM.

The ability of the resulting classifiers—in terms of correctly predicting human physical activity based on accelerometer data—was in terms of consistency and accuracy.

To address the limitations of the proposed architectures, future work will focus on advanced model architectures. We will also more sophisticated model architectures such as attention mechanisms and more advanced types of recurrent networks that would identify subtle temporal patterns which could explain such differences. Besides, sensor fusion will be employed by integrating different sensor data- such as gyroscope, and magnetometer-to

extract comprehensive and discriminative patterns from their fusion, hence attaining the accurate classification needed.

The use of DL in automated activity recognition systems also opens up new research possibilities. By collecting data on human behavior, we can better understand how people interact with their environments and discover new trends. This knowledge can be used to create smarter systems that are better able to predict and respond to user needs. This opens up one of our future perspectives.

**Author Contributions:** Conceptualization, M.S.H., A.H. and M.A.; Methodology, M.S.H. and A.H.; Software, M.S.H. and A.H.; Validation, M.S.H., A.H. and M.A.; Formal analysis, S.A.A. and E.A.; Investigation, E.A.; Resources, S.A.A. and M.A.; Data curation, S.A.A., M.A. and E.A.; Writing—original draft, S.A.A., M.A. and E.A.; Writing—review & editing, A.H.; Visualization, S.A.A. and E.A.; Supervision, M.S.H.; Project administration, M.S.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. Data can be found here: <https://www.cis.fordham.edu/wisdm/dataset.php>; accessed on 30 September 2024 and <https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>; accessed on 1 February 2025.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
DL	Deep Learning
DT	Decision Trees
GPS	Global Positioning System
HAR	Human Activity Recognition
HMM	Hidden Markov Models
k-NN	k-Nearest Neighbors
LSTM	Long Short Term Memory
MEMS	Micro-Electro-Mechanical Systems
ML	Machine Learning
NB	Naive Bayes
ReLU	Rectified Linear Unit
RF	Random Forests
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
WISDM	Wireless Sensor Data Mining

## References

1. Hidri, A.; Mkhinini Gahar, R.; Sassi Hidri, M. What factors distinguish overlapping Data job postings? Towards ML-based models for job category's factors prediction. *Intell. Decis. Technol.* **2024**, *18*, 2161–2176. [[CrossRef](#)]
2. Kollathodi, M.A. A comprehensive comparison and analysis of machine learning algorithms including evaluation optimized for geographic location prediction based on twitter tweets datasets. *Cogent Eng.* **2023**, *10*. [[CrossRef](#)]
3. Madabhushi, A.; Aggarwal, J. A bayesian approach to human activity recognition. In Proceedings of the Second IEEE Workshop on Visual Surveillance (VS), Fort Collins, CO, USA, 26 June 1999; pp. 25–32.

4. Patel, A.T.; Shah, J.H. Performance analysis of supervised machine learning algorithms to recognize human activity in ambient assisted living environment. In Proceedings of the IEEE 16th India Council International Conference (INDICON), Rajkot, India, 13–15 December 2019; pp. 1–4.
5. Popescu, M.C.; Balas, V.E.; Perescu-Popescu, L.; Mastorakis, N. Multilayer perceptron and neural networks. *WSEAS Trans. Circuits Syst.* **2009**, *8*, 579–588.
6. Shen, J.; Fang, H. Human activity recognition using gaussian naive bayes algorithm in smart home. *J. Phys. Conf. Ser.* **2020**, *1631*, 012059. [[CrossRef](#)]
7. Suto, J.; Oniga, S.; Lung, C.; Orha, I. Comparison of offline and real-time human activity recognition results using machine learning techniques. *Neural Comput. Appl.* **2020**, *32*, 15673–15686. [[CrossRef](#)]
8. Ferjani, I.; Sassi Hidri, M.; Frihida, A. SiNoptic: Swarm intelligence optimisation of convolutional neural network architectures for text classification. *Int. J. Comput. Appl. Technol.* **2022**, *68*, 82–100. [[CrossRef](#)]
9. Lee, S.-M.; Yoon, S.M.; Cho, H. Human activity recognition from accelerometer data using convolutional neural network. In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, Republic of Korea, 13–16 February 2017; pp. 131–134.
10. Sassi Hidri, M. Multistep Time Series Forecasting of Energy Consumption Based on Stacked Deep LSTM Network Architecture. In Proceedings of the 16th International Conference on Computational Collective Intelligencerocedia Computer Science (ICCCI), Leipzig, Germany, 9–11 September 2024; pp. 132–143.
11. Babiker, M.; Khalifa, O.; Htike, K.; Hashim, A.; Zaharadeen, M. Automated daily human activity recognition for video surveillance using neural network. In Proceedings of the IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Putrajaya, Malaysia, 28–30 November 2017; pp. 1–5.
12. Liagkou, V.; Sakka, S.; Stylios, C. Security and privacy vulnerabilities in human activity recognition systems. In Proceedings of the 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Ioannina, Greece, 23–25 September 2022; pp. 1–6.
13. Sakka, S.; Liagkou, V.; Stylios, C. Exploiting security issues in human activity recognition systems (HARSS). *Information* **2023**, *14*, 315. [[CrossRef](#)]
14. Singh, D.; Vishwakarma, D. Human Activity Recognition in Video Benchmarks: A Survey. In *Advances in Signal Processing and Communication*; Springer: Singapore, 2019; pp. 247–259.
15. Jalal, A.; Kamal, S.; Kim, D. A depth video-based human detection and activity recognition using multi-features and embedded hidden markov models for health care monitoring systems. *Int. J. Interact. Multim. Artif. Intell.* **2017**, *4*, 54–62. [[CrossRef](#)]
16. Ogbuabor, G.; La, R. Human activity recognition for healthcare using smartphones. In Proceedings of the 10th International Conference on Machine Learning and Computing (ICMLC), Macau, China, 26–28 February 2018; pp. 41–46.
17. Subasi, A.; Radhwan, M.; Kurdi, R.; Khateeb, K. IOT based mobile healthcare system for human activity recognition. In Proceedings of the 15th Learning and Technology Conference (L&T), Gothenburg, Sweden, 28–29 May 2018; pp. 29–34.
18. Zhou, Z.; Yu, H.; Shi, H. Human activity recognition based on improved bayesian convolution network to analyze health care data using wearable iot device. *IEEE Access* **2020**, *8*, 86411–86418. [[CrossRef](#)]
19. Alizadeh, R.; Savaria, Y.; Nerguzian, C. Human activity recognition and people count for a smart public transportation system. In Proceedings of the IEEE 4th 5G World Forum (5GWF), Montreal, QC, Canada, 13–15 October 2021; pp. 182–187.
20. Sekiguchi, R.; Abe, K.; Shogo, S.; Kumano, M.; Asakura, D.; Okabe, R.; Kariya, T.; Kawakatsu, M. Phased human activity recognition based on GPS. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers, Virtual, 21–26 September 2021; pp. 396–400.
21. Xing, Y.; Lv, C.; Wang, H.; Cao, D.; Velenis, E.; Wang, F. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5379–5390. [[CrossRef](#)]
22. Ye, J.; Chen, W.; Li, X.; Zhang, Q.; Zhang, X. Deep learning-based human activity real-time recognition for pedestrian navigation. *Sensors* **2020**, *20*, 2574. [[CrossRef](#)] [[PubMed](#)]
23. Akter, M.; Ansary, S.; Khan, M.A.-M.; Kim, D. Human activity recognition using attention-mechanism-based deep learning feature combination. *Sensors* **2023**, *23*, 5715. [[CrossRef](#)]
24. Bagci Das, D.; Birant, D. Human activity recognition based on multi-instance learning. *Expert Syst.* **2023**, *40*, e13256. [[CrossRef](#)]
25. Dentamaro, V.; Gattulli, V.; Impedovo, D.; Manca, F. Human activity recognition with smartphone-integrated sensors: A survey. *Expert Syst. Appl.* **2024**, *246*, 123143. [[CrossRef](#)]
26. Huang, X.; Zhang, S. Human activity recognition based on transformer in smart home. In Proceedings of the 2nd Asia Conference on Algorithms, Computing and Machine Learning (CACML), Shanghai, China, 17–19 March 2023; pp. 520–525.
27. Morshed, M.G.; Sultana, T.; Alam, A.; Lee, Y.-K. Human action recognition: A taxonomy-based survey, updates, and opportunities. *Sensors* **2023**, *23*, 2182. [[CrossRef](#)] [[PubMed](#)]

28. Parmar, D.; Bhardwaj, M.; Garg, A.; Kapoor, A.; Mishra, A. Human activity recognition system. In Proceedings of the International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 20–21 April 2023; pp. 533–535.
29. Sinha, K.P.; Kumar, P. Human activity recognition from UAV videos using a novel DMLC-CNN model. *Image Vis. Comput.* **2023**, *134*, 104674. [[CrossRef](#)]
30. Zheng, Y.; Wong, W.-K.; Guan, X.; Trost, S. Physical activity recognition from accelerometer data using a multi-scale ensemble method. In Proceedings of the Conference on Innovative Applications of Artificial Intelligence (IAAI), Bellevue, WA, USA, 14–18 July 2013.
31. Surek, G.A.S.; Seman, L.O.; Stefenon, S.F.; Mariani, V.C.; Coelho, L.d.S. Video-based human activity recognition using deep learning approaches. *Sensors* **2023**, *23*, 6384. [[CrossRef](#)]
32. Chen, Y.; Xue, Y. A deep learning approach to human activity recognition based on single accelerometer. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; pp. 1488–1492.
33. Chen, C.; Jafari, R.; Kehtarnavaz, N. A survey of depth and inertial sensor fusion for human action recognition. *Multimed. Tools Appl.* **2017**, *76*, 4405–4425. [[CrossRef](#)]
34. Medrano, C.; Plaza, I.; Igual, R.; Sanchez, A.; Castro, M. The effect of personalization on smartphone-based fall detectors. *Sensors* **2016**, *16*, 117. [[CrossRef](#)]
35. Wang, S.; Zhou, G. A review on radio based activity recognition. *Digit. Commun. Netw.* **2015**, *1*, 20–29. [[CrossRef](#)]
36. Oleh, U.; Obermaisser, R.; Ahammed, A.S. A Review of Recent Techniques for Human Activity Recognition: Multimodality, Reinforcement Learning, and Language Models. *Algorithms* **2024**, *17*, 434. [[CrossRef](#)]
37. Chen, K.; Zhang, D.; Yao, L.; Guo, B.; Yu, Z.; Liu, Y. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Comput. Surv.* **2020**, *54*, 1–40. [[CrossRef](#)]
38. Dua, N.; Singh, S.; Challa, S.; Semwal, V.; Kumar, M. A Survey on Human Activity Recognition Using Deep Learning Techniques and Wearable Sensor Data. In Proceedings of the 4th International Conference on Machine Learning, Image Processing, Network Security and Data Sciences, Virtual Event, 19–20 January 2023; pp. 52–71.
39. Gu, F.; Chung, M.-H.; Chignell, M.; Valaei, S.; Zhou, B.; Liu, X. A survey on deep learning for human activity recognition. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–34. [[CrossRef](#)]
40. Sarveshwaran, V.; Joseph, I.T.; Maravarman, M.; Karthikeyan, P. Investigation on human activity recognition using deep learning. *Procedia Comput. Sci.* **2022**, *204*, 73–80. [[CrossRef](#)]
41. Sousa Lima, W.; Souto, E.; El-Khatib, K.; Jalali, R.; Gama, J. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors* **2019**, *19*, 3213. [[CrossRef](#)]
42. Budisteanu, E.A.; Mocanu, I.G. Combining supervised and unsupervised learning algorithms for human activity recognition. *Sensors* **2021**, *21*, 6309. [[CrossRef](#)]
43. Ma, H.; Zhang, Z.; Li, W.; Lu, S. Unsupervised human activity representation learning with multi-task deep clustering. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–25. [[CrossRef](#)]
44. Zhang, X.; Yi, D.; Behdad, S.; Saxena, S. Unsupervised human activity recognition learning for disassembly tasks. *IEEE Trans. Ind. Inform.* **2024**, *20*, 785–794. [[CrossRef](#)]
45. Chen, Z.; Chao, C.; Zheng, T.; Luo, J.; Xiong, J.; Wang, X. RF-based human activity recognition using signal adapted convolutional neural network. *IEEE Trans. Mob. Comput.* **2021**, *21*, 487–499. [[CrossRef](#)]
46. Mubibya, G.S.; Almhana, J. Improving human activity recognition using ml and wearable sensors. In Proceedings of the IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 165–170.
47. Mohsen, S.; Elkaseer, A.; Scholz, S.G. Human activity recognition using k-nearest neighbor machine learning algorithm. In Proceedings of the 8th International Conference on Sustainable Design and Manufacturing (KES-SDM), Split, Croatia, 15–17 September 2021; pp. 304–313.
48. Wang, P.; Zhang, Y.; Jiang, W. Application of k-nearest neighbor (KNN) algorithm for human action recognition. In Proceedings of the 4th IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 18–20 June 2021; pp. 492–496.
49. Oniga, S.; Suto, J. Human activity recognition using neural networks. In Proceedings of the 2014 15th International Carpathian Control Conference (ICCC), Velke Karlovice, Czech Republic, 28–30 May 2014; pp. 403–406.
50. Kusuma, W.A.; Minarno, A.E.; Safitri, N.D.N. Human activity recognition utilizing svm algorithm with gridsearch. *AIP Conf. Proc.* **2022**, *2453*, 030002.
51. Manosha Chathuramali, K.G.; Rodrigo, R. Faster human activity recognition with SVM. In Proceedings of the International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 12–15 December 2012; pp. 197–203.
52. Qian, H.; Mao, Y.; Xiang, W.; Wang, Z. Recognition of human activities using svm multi-class classifier. *Pattern Recognit. Lett.* **2010**, *31*, 100–111. [[CrossRef](#)]

53. Ariza-Colpas, P.P.; Vicario, E.; Oviedo-Carrascal, A.I.; Butt Aziz, S.; Pieres-Melo, M.A.; Quintero-Linero, A.; Patara, F. Human activity recognition data analysis: History, evolutions, and new trends. *Sensors* **2022**, *22*, 3401. [CrossRef] [PubMed]
54. Kabir, M.H.; Hoque, M.R.; Thapa, K.; Yang, S.H. Two-layer hidden markov model for human activity recognition in home environments. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 4560365. [CrossRef]
55. Manouchehri, N.; Bouguila, N. Human activity recognition with an HMM-based generative model. *Sensors* **2023**, *23*, 1390. [CrossRef]
56. Xu, W.; Pang, Y.; Yang, Y.; Liu, Y. Human activity recognition based on convolutional neural network. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 165–170.
57. Pienaar, S.; Malekian, R. Human activity recognition using LSTM-RNN deep neural network architecture. In Proceedings of the IEEE 2nd Wireless Africa Conference (WAC), Pretoria, South Africa, 18–20 August 2019; pp. 1–5.
58. Yu, T.; Chen, J.; Yan, N.; Liu, X. A multi-layer parallel lstm network for human activity recognition with smartphone sensors. In Proceedings of the 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; pp. 1–6.
59. Mekruksavanich, S.; Jitpattanakul, A. LSTM Networks Using Smartphone Data for Sensor-Based Human Activity Recognition in Smart Homes. *Sensors* **2021**, *21*, 1636. [CrossRef] [PubMed]
60. Dirgová Luptáková, I.; Kubovčík, M.; Pospíchal, J. Wearable Sensor-Based Human Activity Recognition with Transformer Model. *Sensors* **2022**, *22*, 1911. [CrossRef]
61. Srivatsa, P.; Plötz, T. Using Graphs to Perform Effective Sensor-Based Human Activity Recognition in Smart Homes. *Sensors* **2024**, *24*, 3944. [CrossRef]
62. Manjulalayam, R.; Vyas, B.; Patel, R.; Goswami, A.; Mistry, H.; Mavani, C. A Comparative Study of Deep Learning Architectures for Activity Recognition. In Proceedings of the 3rd International Conference on Computational Modelling, Simulation and Optimization (ICCMOSO), Phuket, Thailand, 14–16 June 2024; pp. 380–386.
63. Sheng, M.; Jiang, J.; Su, B.; Tang, Q.; Yahya, A.; Wang, G. Short-time activity recognition with wearable sensors using convolutional neural network. In Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI), Zhuhai, China, 3–4 December 2016; pp. 413–416.
64. Barros, T.; SouzaNeto, P.; Silva, I.; Guedes, L.A. Predictive models for imbalanced data: A school dropout perspective. *Educ. Sci.* **2019**, *9*, 275. [CrossRef]
65. Mohammed, R.; Rawashdeh, J.; Abdullah, M. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In Proceedings of the 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 243–248.
66. Varotto, G.; Susi, G.; Tassi, L.; Gozzo, F.; Franceschetti, S.; Panzica, F. Comparison of resampling techniques for imbalanced datasets in machine learning: Application to epileptogenic zone localization from interictal intracranial EEG recordings in patients with focal epilepsy. *Front. Neuroinform.* **2021**, *15*, 715421. [CrossRef]
67. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
68. Agarap, A.F. Deep learning using rectified linear units (ReLU). *arXiv* **2018**, arXiv:1803.08375.
69. Gulli, A.; Pal, S. *Deep Learning with Keras*; Packt Publishing Ltd.: Birmingham, UK, 2017.
70. Prasad, A.; Tyagi, A.K.; Althobaiti, M.M.; Almulhihi, A.; Mansour, R.F.; Mahmoud, A.M. Human activity recognition using cell phone-based accelerometer and convolutional neural network. *Appl. Sci.* **2021**, *11*, 12099. [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.