International Journal of Computational Intelligence Systems (2024) 17:278
https://doi.org/10.1007/s44196-024-00689-0

RESEARCH ARTICLE

A Novel CNN-BiLSTM-GRU Hybrid Deep Learning Model for Human Activity Recognition

Pooja Lalwani[1] · R. Ganeshan[1]

Received: 23 June 2024 / Accepted: 23 October 2024

Abstract

Human Activity Recognition (HAR) is critical in a variety of disciplines, including healthcare and robotics. This paper presents a new Convolutional Neural Network with Bidirectional Long Short-Term Memory and along with Gated Recurrent Unit (CNN-BiLSTM-GRU)hybrid deep learning model designed for Human Activity Recognition (HAR) that makes use of data from wearable sensors and mobile devices. Surprisingly, the model achieves an amazing accuracy rate of 99.7% on the difficult Wireless Sensor Data Mining (WISDM) dataset, demonstrating its ability to properly identify human behaviors. This study emphasizes parameter optimization, with a focus on batch size 0.3 as a significant component in improving the model's robustness. Furthermore, the findings of this study have far-reaching implications for bipedal robotics, where precise HAR (Human Activity Recognition) is critical to improving human–robot interaction quality and overall work efficiency. These discoveries not only strengthen Human Activity Recognition (HAR) techniques, but also provide practical benefits in real-world applications, particularly in the robotics and healthcare areas. This study thus makes a significant contribution to the continuous development of Human Activity Recognition methods and their actual applications, emphasizing their important role in stimulating innovation and efficiency across a wide range of industries.

Keywords Human activity recognition · Deep learning models · Bipedal robots · Accelerometer · Sensors · Convolutional neural networks · Long short-term memory · Bidirectional long short-term memory · Smartphone

1 Introduction

Predicting human motions via computer interaction in Human Activity Recognition is a complicated and multidimensional challenge despite its numerous applications that considerably improve people's lives. Mobile sensors, such as accelerometers, barometers, and gyroscopes, play an important role in converting physical motion into detectable signals for Human Activity Recognition systems, providing users with both privacy and an alternative method of motion acquisition that avoids the environmental constraints commonly associated with video-based systems. However, despite advances in sensor technology, there are inherent limits in effectively recognizing and interpreting human activities, particularly in terms of a thorough grasp of various stages of humanoid body position. This constraint may lead to performance loss, especially in scenarios demanding accurate gesture capture and presentation, necessitating the use of many contributing sensors in industrial contexts to increase overall system efficacy and dependability [1].

HAR is currently used in a broad range of applications, including smart home functionality, healthcare, rehabilitation, and intelligent surveillance systems, as well as in-depth gait and gesture analysis [2–5]. At its foundation, Human Activity Recognition seeks to understand and anticipate human intentions and behaviors utilizing a variety of sensor technologies customized to certain environmental circumstances. These diverse data sensory systems encompass a spectrum of technologies, including accelerometers, smartphones, Wi-Fi signals, depth cameras, infrared sensors, audio inputs, and Bluetooth, each offering unique advantages and insights into human activity patterns based on their application contexts [6].

✉ Pooja Lalwani
poojalalwani2020@vitbhopal.ac.in

R. Ganeshan
ganeshramasamy111@gmail.com

1 School of Computing Science and Engineering, VIT Bhopal University, Kothri Kalan, Astha, Sehore 466114, Madhya Pradesh, India

Because of their responsiveness to changes in human gaits and motions, mobile sensors' inherent features, such as their ability to capture acceleration and rotational velocity, offer considerable potential for inferring human activities. Furthermore, the small and adaptable nature of mobile sensors enables for their incorporation into a wide range of devices and places, resulting in cost-effectiveness, energy efficiency, robust performance, and less dependency on external environmental variables. Activity detection systems employing smartphone sensors are gaining a lot of attention due to their increasing prevalence in daily activities. To precisely recognize human movement, researchers are concentrating on determining the best sensor designs [7].

Deep Learning (DL) algorithms have become a dominant force in diverse industries, demonstrating outstanding performance in tasks, such as object detection [8], classification [9], and natural language processing [10]. These deep learning frameworks, distinguished by their ability to extract complex characteristics from large datasets autonomously, have transformed the landscape of Human Activity Recognition (HAR) approaches. A study using a tri-axial accelerometer demonstrated the efficiency of deep learning (DL) methods in forecasting arm movements using convolutional neural networks. This emphasizes the significance of feature extraction methods like Fourier and wavelet transforms, statistical measures, and a range of Machine Learning models, such as K-Nearest Neighbor, Random Forest, Decision Trees, and neural networks [11].

Despite the tremendous advances made by Deep Learning in Human Activity Recognition, obstacles remain, particularly in the field of feature extraction due to issues like as noisy data and class imbalances. Many Human Activity Recognition(HAR) techniques continue to rely significantly on human feature engineering, underscoring the continuous need for novel ways to expedite data pre-processing and improve raw sensor data utilization. This paper presents a unique Deep Learning model (DL) designed to process raw sensor input directly, eliminating the complications inherent with existing feature extraction approaches and providing a more simplified approach to HAR system development.

This article describes a Human Activity Recognition (HAR) technique that leverages wearable sensor data using numerous branches of the Convolutional Neural Network-Bidirectional Gated Recurrent Unit model with different kernel sizes. This novel approach combines CNNs' strengths in extracting local features with Bidirectional Gated Recurrent Units (BiGRU) ability to capture long-term relationships, resulting in improved prediction accuracy and overall system performance. Furthermore, by incorporating convolutional kernels of various sizes, the model effectively captures the intricate temporal correlations inherent in sequential data, as demonstrated by its robustness and efficacy on proven datasets such as Wireless Sensor Data Mining (WISDM) for evaluating human behavior comprehension.

1.1 Author Contributions

- In this work, wearable sensor data is used to present a novel hybrid method to human activity recognition. With minimum pre-processing, the model runs directly on raw data by combining multi-branch Convolutional Neural Network-Bidirectional Long Short Term Memory-Gated Recurrent Unit components.
- The Convolutional Neural Network-Bidirectional Long Short Term Memory-Gated Recurrent Unit model outperformed the other models, according to the comparison. It is a three-branch model that uses different filter sizes (0.2, 0.3, and 0.4, for example). The Recurrent Neural Network layer successfully preserves long-term dependency in the data by mixing several filter sizes. After the Recurrent Neural Network layer, a fully linked layer with a softmax activation function makes it simple to generate an abstract representation of the input data.
- Convolutional Neural Network, Gated Recurrent Unit, and Bidirectional Long Short-Term Memory networks are used by the model to efficiently capture both short- and long-term associations in sequential data.
- The new strategy used by the model to combine different filter sizes improves feature extraction capabilities, which increases the ability to capture a variety of temporal and spatial correlations.
- Two segments of the dataset were created for training and testing after pre-processing. 20% of the data were sent to the testing set and the remaining 80% to the training set.
- When tested, notably on publically available datasets such as Wireless Sensor Data Mining, our model performed better than other previous Deep Learning methods for Human Activity Recognition that have been reported in the literature.

1.2 Structuring the Article

The article is divided into six sections: an introduction, related work, methodology, proposed methodology, experiments and results, and conclusions and future findings.

2 Literature Review

Deep Learning relies largely on Recurrent Neural Networks (RNNs) to deal with sequential input, notably for capturing long-term dependencies in data [12].

Human Activity Recognition (HAR) aims to classify and detect common human motions as jogging, walking, standing, sitting, and going downstairs and upstairs. The best classifier for achieving the best recognition performance must be chosen through comparative analysis with 96.81% accuracy and a mean error of 0.03 on publicly available Human Activity Recognition (HAR) datasets, Bozkurt et al. [13] discovered that Machine Learning and Deep Learning models performed admirably. The authors Dua et al. [14] achieved accuracies of 97.21%, 95.27%, and 96.20% on various datasets through the usage of Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) for feature extraction and classification. Guan et al. [15] developed a unique training strategy for Long Short-Term Memory models coupled with an ensemble classifier, and the results were encouraging. Deep learning models have demonstrated the benefits of ensemble learning techniques, which are frequently employed in machine learning. An accurate learning preference is maintained while user data is preserved, thanks to a privacy-protected federated custom random forest structure established by Liu et al. [16] for tailored human activity identification.

A Bidirectional Long Short-Term Memory (BiLSTM)-based network for Human Activity Recognition (HAR) was constructed by Hernandez et al. (2019) using smartphone sensor data. Through the parallel fusion of an Long Short-Term Memory (LSTM with a fully convolutional block, Karim et al. [17] developed a hybrid Human Activity Recognition (HAR) model. By integrating the best features from each viewpoint, this innovative mix sought to improve the model's overall performance. This improved the model's temporal and spatial dimensions and raised the identification rate. Using spatiotemporal sequence forecasting, Mutegeki et al. (2020) improved the prediction accuracy of human activities derived from unprocessed sensor data utilizing long short-term memory (LSTM) and convolutional neural networks (CNNs). The problems with multimodal wearable sensors were tackled by Ordonez et al. [18] using a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) architecture tuned for Human Activity Recognition (HAR). As a Recurrent Neural Network (RNN) variation, Long Short-Term Memory (LSTM) addresses the vanishing gradient issue that regular Recurrent Neural Network (RNNs) have by managing long-term dependencies well. Ullah et al. (2019) used smartphone sensor data to recognize different human behaviors by stacking five Long Short-Term Memory (LSTM) cells to create a strong classifier. [19], also presented a deep residual bidirectional Long Short-Term Memory (LSTM) architecture that establishes bidirectional connections with forward and backward states attached.

Using a Convolutional Neural Network (CNN) framework and smartphone sensors, researchers in [20] developed a Human Activity Recognition (HAR) model to recognize different human behaviors. A unique deep learning framework for real-time human activity classification was presented in [21], which included Convolutional Neural Network (CNN) and statistical characteristics. For Human Activity Recognition (HAR), attention-based techniques were introduced in [22]. These techniques use tagged activity data from wearable sensors to combine Convolutional Neural Network (CNN) algorithms with attention sub-modules to improve feature correlations. Teng et al. [23] presented a new architecture for activity recognition that replaces the global loss of the baseline Convolutional Neural Network (CNN) with a local loss in an effort to minimize the amount of memory required for sensor-based recognition. A Convolutional Neural Network (CNN)-driven Human Activity Recognition (HAR) approach is used to capture scale invariance and local dependency in data from mobile sensors [24].

According to [25], this study includes a range of Deep Learning (DL) models for Human Activity Recognition (HAR) and feature extraction. First, theConvolutional Neural Network (CNN) model is used to determine the effectiveness of the Human Activity Recognition (HAR) design. Second, certain intricate models like Recurrent Neural Network (RNN) and self-attention are combined into the sub-modules. The author conducted experiments to demonstrate that the origin of V3, which is utilized to execute cross-channel multi-size convolution transformation, performs better than other spines using publically available datasets including Human Activity Sensing Consortium (HASC), Wireless Sensor Data Mining (WISDM), and University of California,Irvine (UCI). According to the author, SeNet performs better than other sub-modules. Thakur et al. (2022) developed deep learning techniques for human movement identification using accelerometer and gyroscope sensor data from smartphones. Long Short-Term Memory is utilized for temporal modeling, AEs (Autoencoders) for dimension reduction, and Convolutional Neural Network (CNN) for feature extraction. Utilizing three distinct models Convolutional Neural Network, Autoencoders, Long Term Short Term (CNN, AEs, and LSTM), the proposed architecture, named "ConvAE-LSTM," was developed and evaluated across multiple datasets, such as University of California, Irvine (UCI), Physical Activity Monitoring (PAMAP2), Wireless Sensor Data Mining(WISDM), and OPPORTUNITY. This enhances, among other things, the computational efficiency, precision, recall, and accuracy of the intriguing processes. According to the study, Human Activity Recognition (HAR) can be used as a model to identify human movements and activities [26]. The author tracks people's behavior using a variety of devices, such as a smartwatch and a smartphone. Many classification methods, including Random Forest, Simple Logistic, and Sequential Minimal optimization (SMO), were used. Utilizing the University of California, Irvine (UCI)-Human Activity Recognition (UCI-HAR) and Wireless Sensor Data Mining (WISDM) datasets,

Table 1 The state-of-art of existing work

Author name/References	Title	Description of dataset	Models	Accuracy
Yasin kaya et.al, [27], 2024	Using Deep CNN to Recognise Human Activity from Data from Multiple Sensors	PAMAP2 Dataset, WISDM Dataset	1D-CNN	90.27%, 97.8%
Shanying Zhu [28], 2023	Human Activity Recognition Based on a modified capsule network	WISDM Dataset, UCI-HAR Dataset	CNN	91.62%. 96.08%
Ferhat Bozkurt et al. [13], 2022	A comparative study on classifying human activities using classical machine and deep learning methods	UCI-HAR Dataset	DNN, SVM+KNN	96.71%, 96.81%
Md. Milon Islam et al. [29], 2022	Human activity recognition using tools of convolutional neural networks: A state of the art review	Vision Dataset, Image Dataset, HAR Dataset	CNN	–
Songfeng, Liu et al. [16], 2022	Federated personalized random forest for human activity recognition	WISDM Dataset, UCI-Smartphone Dataset	PP-FPRF	94.5% 93.0%
Nayak Suvra et al. [26], 2022	Comparative analysis of HAR datasets using classification algorithms	WISDM Dataset, UCI-HAR Dataset	SMO,RF,SL	90.69%, 98%
Yin Tang et al. [30], 2022	Triple cross-domain attention on human activity recognition using wearable sensors	UNIMIB-SHAR Dataset, UCI-HAR Dataset, WISDM Dataset, PAMAP2 Dataset	ResNet	78.55%, 96.77% 98.61% 96.77%
Dipanwita Thakur et al. [31], 2022	ConAE-LSTM: Convolutional auto- encoder long short-term memory network for smartphone-based human activity recognition	PAMAP2 Dataset, WISDM Dataset, UCI-HAR Dataset	ConAE-LSTM	94.33%, 98.67%, 97.13%
Z Zhongkai et al. [25], 2022	Toward an effective cnvolutional neural network architecture for sensor-based human activity recognition	HASC Dataset, WISDM Dataset, UCI-Smartphone Dataset	Inception V3, DenseNet121	92.55%, 91.54%, 94.23%
Ahmed Mohamed Helmi et al. [32], 2021	A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human activity recognition using smartphone sensors entropy	WISDM Dataset, UCI-HAR Dataset	GBOGWO	98%
Nidhi Dua et al. [14], 2021	Multi-input CNN-GRU based human activity recognition using wearable sensors	PAMAP2 Dataset, WISDM Dataset, UCI-HAR Dataset	CNN-GRU	95.27%, 97.21%, 96.20%
Kishor Walse et al. [33], 2016	A study of human activity recognition using Adaboost classifiers on WISDM dataset	WISDM Dataset	RF,REP Tree	94.60%

they obtained 98% and 90.69% of the outcomes, respectively. Table 1 provides a summary of the literature review results.

3 Methodology

The following subsections go into great length on the methodology, including a description of the full dataset, a comprehensive overview of data pre-treatment techniques, and a detailed examination of various deep learning models.

3.1 Description of Wireless Sensor Data Mining(WISDM) Dataset

This study made use of a Human Activity Recognition (HAR) dataset from the Wireless Sensor Data Mining(WISDM) group that is available to the public [33]. This dataset was acquired through the use of an Android mobile application. The participants were told to put their phones in their front leg pockets and perform five different activities-walking, jogging, climbing and descending stairs, sitting, and standing-all under close observation. The accelerometer sensor recorded precise time series data by maintaining a sampling rate of 20 Hz during these operations. The table of the Wireless Sensor Data Mining (WISDM) Human Activity Recognition (HAR) dataset provides an in-depth analysis of the raw accelerometer measurements, and Table 2 provides a thorough tabular representation of the entire dataset.

3.2 Data Pre-Processing

The upcoming subsections delve into a detailed description of the data pre-processing techniques. These techniques encompass Linear Interpolation, Scaling, and Normalization, Segmentation, and Automatic Feature Extraction using 1-D Convolutional Neural Networks. Subsequent sections offer additional insights into the conducted tests' analysis and the conclusions derived from the results.

3.2.1 Linear Interpolation

The datasets under analysis are real-world sources of data acquired on humans via wearable wireless sensors. This means that there's a chance of data loss throughout the gathering process, which is often represented by NaN or 0 values. The study used the linear interpolation method, which fills in gaps by predicting intermediate values based on surrounding data points, to lessen the impact of missing data points and address this problem.

3.2.2 Scaling and Normalization

The significance of normalizing input data between 0 and 1 is emphasized by Equation (1). In particular, when working with massive amounts of data that are directly received from multiple channels, this normalization is essential since it helps reduce biases that could arise during model training.

$$M_{\mathrm{l}} = \frac{M_l - \mathrm{m}_{l\,\min}}{\mathrm{m}_{l\,\max} - \mathrm{m}_{l\,\min}} \quad (l = 1, 2, \cdots, p) \tag{1}$$

Here, mlmax and mlmin represent the highest and lowest values of the lth channel, respectively, while p denotes the total number of channels.

3.2.3 Segmentation

Constant signals are produced by wearable sensors and smartphones, frequently in the form of time series data that records action sequences across time. This sensor data must be separated as the first step in the detection of human activity. Sliding windows, which divide the sensor data into fixed-size windows, are a popular technique for this separation. This study's suggested model made use of a sliding window with a size of 128 (holding 128 readings per window) and a 50% overlap over all datasets.

3.2.4 Feature Extraction

Time series data from cellphones and wearable sensors is essential for comprehending behavior in people. High cross-temporal correlations between neighboring variables and a clear one-dimensional structure were noted by LeCun et al. (1995) in this kind of data. Because convolutional neural networks (CNNs) may use tiny receptive fields, they are particularly good at capturing local details. CNNs are used to extract features from time series data, as shown graphically in Fig. 1. Convolutional Neural Network (CNNs) are used to analyze time series data with dimensions of (L x N), where L is the data's length and N is its number of channels (1-N). Convolutional Neural Network (CNNs) produce feature maps utilizing convolution filters, which streamlines the feature extraction procedure. The Convolutional Neural Network (CNNs) architecture's number of filters and the quantity of generated

3.3 Deep Learning Models Description

The following are some of the deep learning classification models that are covered in this section:

Table 2 Description of dataset performed by human activity recognition

Number of classes	Description of activities	Total number of samples	Percentage records
1	Standing	48,395	4.4%
2	Sitting	59,939	5.5%
3	Walking downstairs	1,00,427	9.1%
4	Walking upstairs	1,22,869	11.2%
5	Jogging	3,42,177	31.2%
6	Walking	4,24,400	38.6%
Total		10,98,207	100%

3.3.1 Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are a class of artificial neural networks distinguished by its deep feedforward design, which consists of several layers with a large number of neurons in each layer. Convolutional, pooling, and fully connected layers are important layers in a Convolutional Neural Network (CNN), and each has a distinct purpose. The six layers that make up the Convolutional Neural Network (CNN) model that is suggested in this study are shown in Fig. 1: two convolutional layers, two pooling layers, one fully connected layer, one input layer, and one output layer. The input layer of this study is in charge of receiving and pre-processing the time series data that is used for experimentation. The job of convolutional layers is to extract features; increasingly intricate and important features are extracted by each subsequent layer. Using max pooling in this instance, the pooling layer seeks to minimize the amount of features. To create an intermediate neural network layer, each convolutional layer is linked with a matching pooling layer. To get the final class scores, the outputs from the convolutional and pooling layers before them are aggregated by three fully connected layers. The output layer produces findings that result in categorization outcomes based on the outputs of the fully connected layers.

Convolutional Neural Networks (CNNs) are distinguished by their unique architecture, which consists of pooling and sequential convolution at every level. Discrete feature vectors can be created by storing different features of input signals using these procedures. After convolutional layers, the popular max pooling layer reduces the amount of data by choosing the maximum value to combine smaller features into a single output.

The output is flattened into a one-dimensional vector for classification following multiple convolutional and max pooling layers. This vector can be modified to include more features as needed. By recognizing nonlinear dependencies, one or more fully linked layers-in addition to Convolutional Neural Network (CNN) layers-help solve classification difficulties. The softmax layer receives the output from the previous layer and uses it to create a probability distribution with the anticipated classes [34].

3.3.2 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) units equipped with recurrent neural networks (RNNs) are commonly used to simulate time series data. The cell state, represented by the horizontal lines at the top of the diagram, is the central component of an Long Short-Term Memory (LSTM). This cell state acts like a conveyor belt, allowing data to move through the process with little interruption and minimum linear exchanges. A gate mechanism that has been painstakingly built for informed decision-making regulates the controlled flow of information into and out of the cell state. This gate allows for fine control over the retention or removal of information by regulating access to the cell state through the use of pointwise multiplication and a sigmoid neural network layer. The ability of Long Short-Term Memory (LSTM) to solve the vanishing gradient problem is one of its main benefits. The weight of the self-loop is dynamically changed in response to changing input, forgetting, and output gate thresholds. The network may efficiently scale across several time steps with consistent model parameters, thanks to this dynamic regulation, which also improves training stability and reduces the problems caused by vanishing or inflating gradients. Many applications in a variety of technological disciplines, including robotics control, image analysis, language translation, document summarization, speech recognition, picture recognition, and handwriting identification, show how versatile LSTM is (Wan 2020 Deep). The architecture of an Long Short-Term Memory (LSTM) is shown in Fig. 2, which also highlights the complex gate structure and information flow dynamics of the device.

3.3.3 Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional LSTM (BiLSTM), an Long Short-Term Memory (LSTM) model extension, employs a dual-calculating technique to generate both forward and backward predictions. Unlike ordinary Long Short-Term Memory (LSTM),

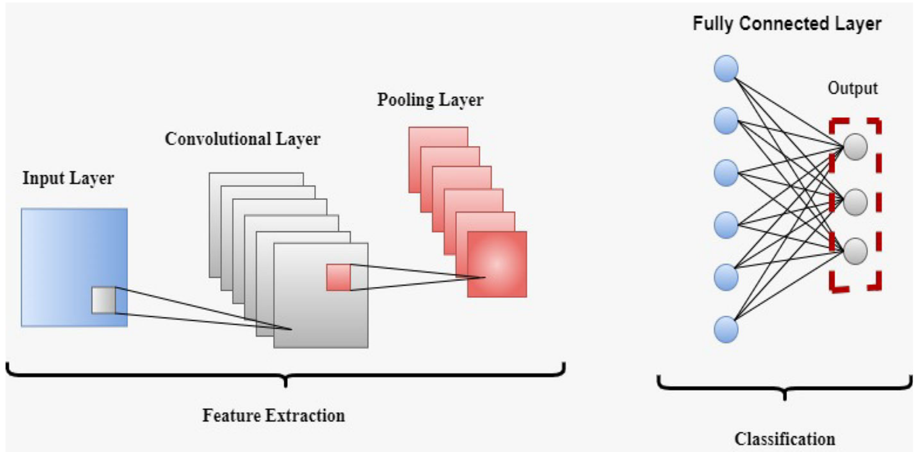


Fig. 1 Convolutional Neural Network Architecture

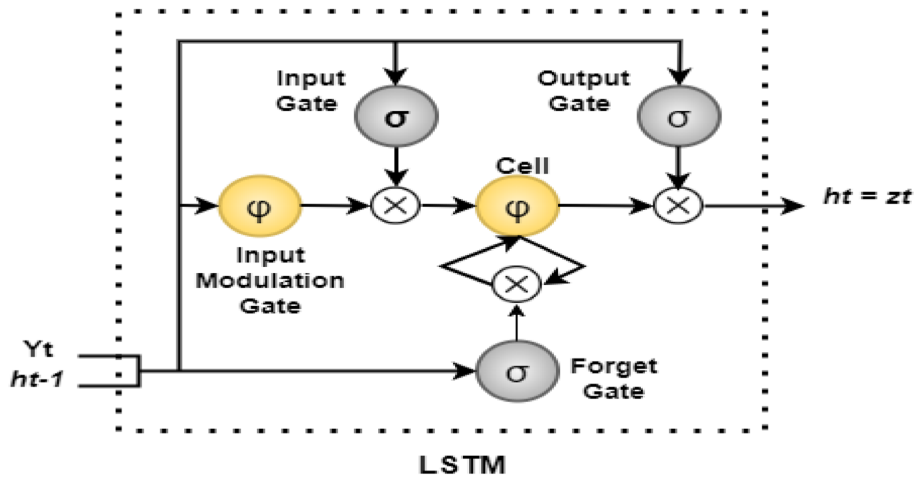


Fig. 2 The Architecture of Long Short-Term memory(LSTM)

which predicts future units only based on past units, Bidirectional Long Short-Term Memory (BiLSTM) accepts input from both sides simultaneously. This is performed utilizing two interconnected Recurrent Neural Network(RNN) structures: one processes historical data forward and the other processes future data backward. Bidirectional Long Short-Term Memory (BiLSTM) accurately captures comprehensive temporal linkages by incorporating information from both previous and future contexts. Bidirectional Long Short-Term Memory (BiLSTM) is extremely effective in situations where there is a significant bidirectional information dependency. Its ability to absorb inputs from both directions improves its performance over unidirectional Long Short-Term Memory (LSTMs). Figure 3 shows Bidirectional Long Short-Term Memory (BiLSTM's) architecture and dual-processing approach.

3.3.4 Gated Recurrent Unit (GRU)

Recurrent Neural Network (RNN) hidden layer employs feedback Loops, which allow them to handle sequential input and excel at tasks like as language modeling and audio recognition. However, Recurrent Neural Networks (RNNs) fail to maintain long-term dependencies when there are large temporal gaps in sequences, which is typically caused by issues like the vanishing gradient problem. To alleviate these limits, architectures like as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) are recommended.

Cho and colleagues introduced the Gated Recurrent Unit (GRU), which solves the vanishing gradient problem in Recurrent Neural Network (RNN). Gated Recurrent Unit (GRU) aims to improve long-term dependence learning. GRU design combines the input and update gates into a single update gate, as well as integrating the hidden and cell states, resulting in a more computationally efficient model than Long Short-Term Memory (LSTM). Gated Recurrent Unit (GRU) efficiently addresses the vanishing gradient problem by selecting relevant data for predictions while excluding irrelevant data. The update gate controls the retention of past data for subsequent stages, whereas the reset gate controls the erasure of previous data. The reset gate is essential for retaining useful knowledge from the past while preserving relevant context. GRU's gate working mechanism enhances its ability to successfully learn and preserve long-term depen-

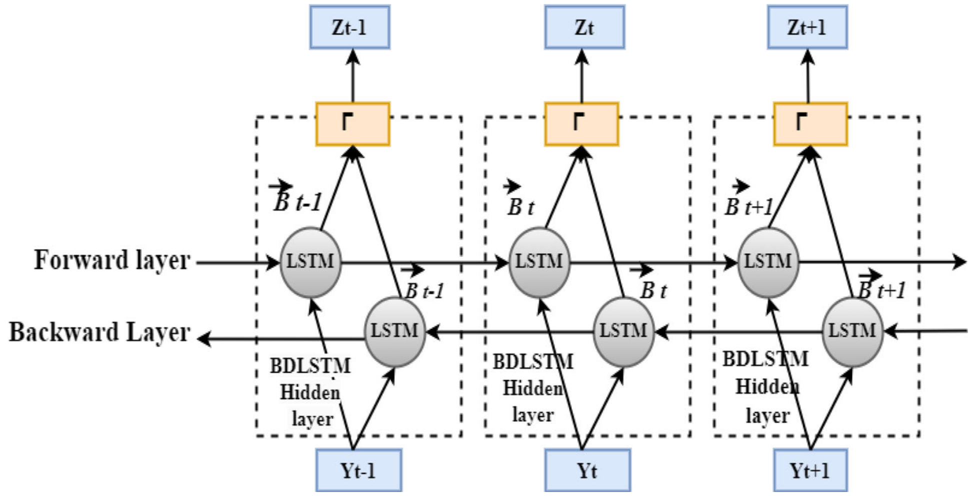


Fig. 3 The Architecture of Bidirectional long short-term memory(BiLSTM)

dencies [35].

$$\text{Update gate: } y_d = \sigma\left(V_y s_d + Q_y l_{d-1}\right) + a_y, \quad (2)$$

$$\text{reset gate: } j_d = \sigma\left(V_j s_d + Q_j l_{d-1}\right) + a_j \quad (3)$$

$$\text{current memory state: } l_d = \tanh\left(V_l s_d + Q_l l_{d-1}\right) + a_l, \quad (4)$$

$$\text{final memory: } l_d = y_d \star l_{d-1} + (1 - y_d) \star l_d, \quad (5)$$

where $\star$ symbolizes the Hadamard product. l_{d-1} refers to the previous state; V_y, V_j, and The weight matrices of the update gate, reset gate, and current memory state are denoted by V_l. The hidden weight matrices of the update gate, reset gate, and current memory state are denoted by Q_y, Q_j, and Q_l, respectively. The bias vectors of the update gate, reset gate, and current memory state are denoted by a_y, a_j, and a_l, respectively. The input feature vector is denoted by y_d, and the sigmoid activation memory state is denoted by σ. Figure 4 depicts the architecture of a GRU cell under examination.

4 Proposed Work

In our proposed methodology, we proposed a Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) hybrid model with a sequential structure. The model begins with a Convolutional Neural Network (CNN) layer, followed by a Maxpooling operation, and lastly a dropout layer. Each transformed feature vector is treated as an input feature. The most widely used pooling method is the max pooling layer, which comes after the convolutional layer. It collects a variety of minor attributes and selects the most valuable one, considerably reducing data dimensionality. Following the first Maxpooling layer, we construct Bidirectional LSTMs (BiLSTM), which are sequence processing models made up of two Long Short-Term Memory (LSTMs) that process input in both forward and backward directions. Following that, a Bidirectional LSTM (BiLSTM) layer is applied to capture long-term, bidirectional relationships between successive data time steps. Following the BiLSTM layer, a Maxpooling layer is used to extract the highest value from the feature mapping based on filter size and stride. To prevent overfitting of the neural network, dropout layers are included after the Maxpooling layer. The GRU (Gated Recurrent Unit) layer, which simulates sequential data by selectively retaining or forgetting information over time and randomly discarding a group of neurons, comes after the third Maxpooling and Dropout layers. Following that, an RNN (Recurrent Neural Network) layer is added, which is designed to handle sequential data like time series, audio, and text. Finally, the final layer's output is processed using a softmax activation function, which normalizes the network's output into a probability distribution that may be used to predict the final result. Tables 3, 4, and 5 offer detailed information about each layer's parameters. Furthermore, it is built with variable branch sizes, such as 0.2, 0.3, 0.4, and so on, and it provides a comprehensive analysis of the setup for various areas of the model. The flowchart for our suggested model depicts the methodical methods and operations that it performs. It displays a graphical representation of the data translation and processing that takes place along the model pipeline. This flowchart depicts the model's structure and the order of actions required, which include data pre-processing, feature extraction, model training, and evaluation. Figure 5 depicts a flowchart that serves as a pathway for understanding the architecture and distinctive components of our suggested model. These experiments show with three different branch width combinations in deep learning models. These branches come in various diameters, such as 0.3, 0.2, and 0.4. They are followed by the hybrid model, which combines the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent

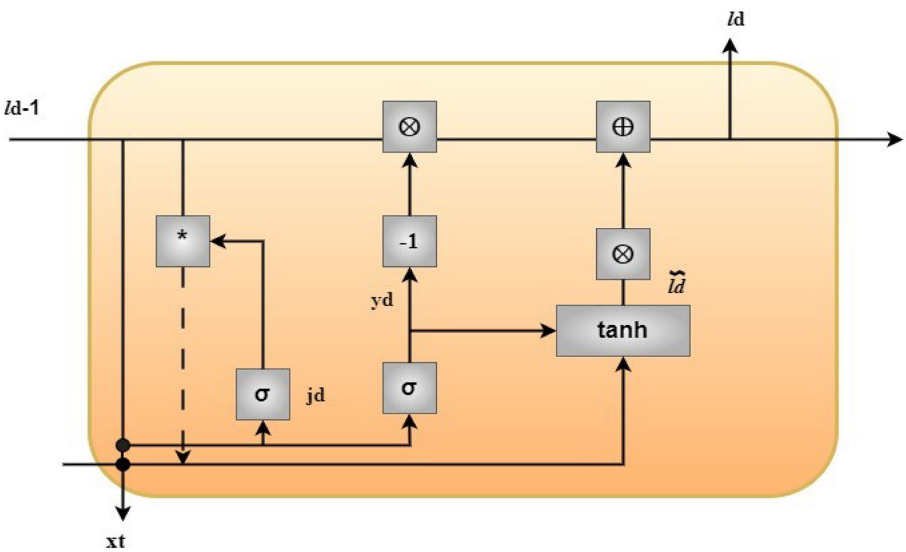


Fig. 4 Architecture Gated Recurrent Unit

Unit (CNN-BiLSTM-GRU) models, as well as Maxpooling and dropout layers. Following that, an RNN layer concatenates and processes the outputs of all three branch sizes. The Dense layer, also known as the fully connected layer, comes after the RNN layer and is responsible for creating an abstract representation of the input data. This fully connected layer is followed by a softmax activation function, which normalizes the network's output to provide a predicted output. Using these numerous options, we may examine and compare the performance of various architectural configurations for our planned study.

5 Experimentation and Results

This section contains discussion, data analysis, performance measurements, and simulated environment comments. The ensuing subsections offer more in-depth analyses of these components.

5.1 The Simulation Environment and Feature Extraction Process

The Python simulations were performed on a PC running Microsoft Windows 11 and equipped with an Intel Core i7 CPU at 3.40 GHz, 16GB of RAM, and a chipset 2600.

5.2 Performance Matrix

In this experimental study, algorithmic performance is evaluated using several metrics, including precision, recall, accuracy, and the F1 score. Furthermore, confusion matrices and accuracy/loss assessments are used to investigate the categorization problems discovered during the experiment. These critical parameters play an important role in determining categorization accuracy.

- **Accuracy** - A classifier's accuracy is determined by dividing the number of successfully classified samples by the total number of samples in a test set. This measure provides information on the percentage of correctly identified samples. The formula to calculate accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

True Positives, True Negatives, False Positives, and False Negatives are essential components of the accuracy formula. These factors, which represent diverse outcomes in classification, are critical for determining overall accuracy.

- **Precision** - Precision is a statistical measure that compares the number of correctly identified positive samples to the total number of positive samples in the identified set. The formula for computing precision is as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

- **Sensitivity** -The ratio of correctly detected positive samples to the total number of positive samples in the dataset is known as the recall rate, also known as the sensitivity or true positive rate. The percentage of positive samples that the model accurately identified is indicated. The following formula can be used to determine the recall rate:

$$\text{Precision} = \frac{TP}{TP + FN} \tag{8}$$

Table 3 Model parameter description of branch-I

Model parameter description of branch_Size = 0.2						
Models	Filters_size	Kernel_size	Return_Sequence	Activation _function	Pool_size	Dropout
CNN-1D maxpooling dropout	128	4	–	Softmax	2	0.2
BiLSTM maxpooling dropout	64	–	True	Softmax	2	0.2
GRU maxpooling dropout	32	–	True	Softmax	2	0.2
LSTM	16	–	–	Softmax	–	0.2
Dense_Layer	6					
Total_Params	119,312					

Table 4 ModeL parameter description of branch-II

Model parameter description of branch_Size = 0.3						
Models	Filters_size	Kernel_size	Return_Sequence	Activation _function	Pool_size	Dropout
CNN-1D maxpooling dropout	256	4	–	Softmax	2	0.3
BiLSTM maxpooling dropout	128	–	True	Softmax	2	0.3
GRU maxpooling dropout	64	–	True	Softmax	2	0.3
LSTM	32	–	–	Softmax	–	0.3
Dense_Layer	6					
Total_Params	472,048					

Table 5 Model parameter description of branch-III

Model parameter description of branch_Size = 0.4						
Models	Filters_size	Kernel_size	Return_Sequence	Activation _function	Pool_size	Dropout
CNN-1D maxpooling dropout	128	4	–	Softmax	2	0.4
BiLSTM maxpooling dropout	128	–	True	Softmax	2	0.4
GRU maxpooling dropout	64	–	True	Softmax	2	0.4
LSTM	32	–	–	Softmax	–	0.4
Dense_Layer	6					
Total_Params	339,312					

- **F1_Score** - The harmonic mean of precision and recall is determined via a statistic called the F1 Score, sometimes referred to as the balanced F1 score. It gives a fair evaluation by taking recall and precision into account. This is the formula used to calculate the F1 Score:

$$F1\_\text{ Score } = \frac{2 \times \text{ Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{9}$$

- **Calculating Confusion Matrix** - The confusion matrix is a square matrix that shows a classification model's overall performance graphically. Its columns show anticipated class labels, and its rows reflect actual class labels. The diagonal elements of the matrix represent the percentage of data points-i.e., occurrences of each class that were properly detected-where the expected and actual labels agree.

5.3 Discussion and Analysis of the Results

We used the data from the following subsections to create convergence curves, evaluate accuracy metrics, and produce a confusion matrix in this section.

5.3.1 Confusion Matrix Calculation for the Suggested Model with Various Batch Sizes

The confusion matrices for the proposed models provide a detailed breakdown of both the model's predictions and the actual class labels in the testing dataset. Each row displays the actual class labels, while each column shows the anticipated class labels. The matrix calculates True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) for each class to provide a complete picture of the model's performance.

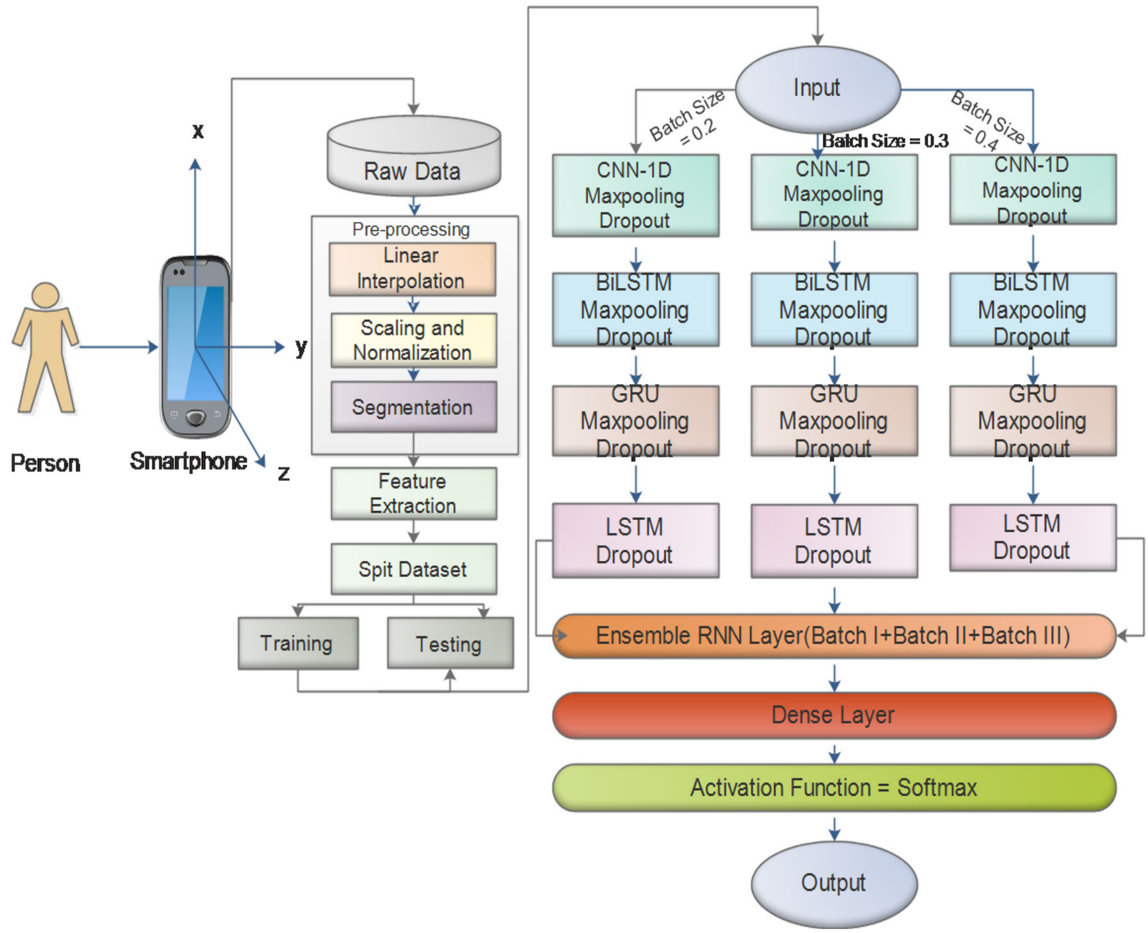


Fig. 5 Flowchart of Proposed Model

The confusion matrix enhances the computation of essential performance indicators like as precision, accuracy, recall, and F1 scores. This strategy ensures a comprehensive and diverse assessment of the model's effectiveness. Our article employs a hybrid deep learning model that blends Convolutional Neural Networks (CNNs) with Bidirectional Long Short-Term Memory networks (BiLSTM) and Gated Recurrent Unit networks (GRU). The model was tested with three different batch sizes: 0.2, 0.3, and 0.4. Using the confusion matrix, we found that the 0.3 batch size consistently outperformed the other batch sizes, resulting in better model performance.

5.3.2 Model Accuracy Evaluation Using the Confusion Matrix

The model's effectiveness was assessed by a comprehensive accuracy assessment over a wide range of activities. This evaluation tested the model's ability to predict the proper class for each instance, necessitating a thorough investigation of the accuracy associated with each activity. The findings of this extensive investigation provide fascinating insights into the model's overall capacity to effectively categorize a wide range of activities. The findings in Tables 6, 7, and 8 were presented using a variety of metrics, including F1 score, Recall, Precision, Macro-average, Weighted Average, and overall accuracy, providing a comprehensive assessment of the model's performance across all classes. The Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) architecture performed admirably with multiple batch sizes (0.2, 0.3, and 0.4), demonstrating the model's versatility across different batches. Notably, batch size 0.4 was the standout performer, outperforming its peers. The total accuracy was assessed to be 93%, 99.7%, and 92.28% for batch sizes 0.2, 0.3, and 0.4, respectively, highlighting the model's consistent proficiency. Additional measurements revealed that the macro-average precision was 92.0%, 92.0%, and 91.0% for batch sizes 0.2, 0.3, and 0.4, respectively. Concurrently, recall values were 91.0%, 93.0%, and 90.0%, demonstrating the model's capacity to effectively capture instances of each class. The F1 Score, which is a compos-

Table 6 Comparative results based on batch size = 0.2

Class	Precision	Recall	F1 Score	Overall accuracy
0	0.92	0.67	0.78	0.93
1	0.98	0.98	0.98	
2	1	0.95	0.97	
3	0.94	0.98	0.96	
4	0.70	0.91	0.79	
5	0.98	0.96	0.97	
Macro-average	0.92	0.91	0.91	
Weighted average	0.94	0.93	0.94	

Table 7 Comparative results based on batch size = 0.3

Class	Precision	Recall	F1 Score	Overall Accuracy
0	0.78	0.89	0.83	0.997
1	0.99	0.97	0.98	
2	0.99	0.95	0.97	
3	0.93	0.99	0.96	
4	0.83	0.83	0.83	
5	0.99	0.97	0.98	
Macro-average	0.92	0.93	0.93	
Weighted average	0.989	0.99	0.99	

Table 8 Comparative results based on batch size = 0.4

Class	Precision	Recall	F1 Score	Overall accuracy
0	0.88	0.68	0.77	0.9228
1	0.99	0.96	0.97	
2	0.99	0.95	0.97	
3	0.94	0.99	0.96	
4	0.68	0.88	0.77	
5	0.97	0.96	0.97	
Macro-average	0.91	0.90	0.90	
Weighted average	0.92	0.93	0.93	

ite measure of precision and recall, was estimated at 91%, 93%, and 90% for all batch sizes, highlighting the model's balanced performance. Surprisingly, the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model achieved its peak when deployed with a batch size of 0.4, yielding the most amazing results across all tested criteria. This highlights the importance of batch size selection in optimizing the model's prediction skills for a variety of activities.

Figures 6, 7, and 8 depict graphical representations of the data presented in Tables 6, 7, and 8. These visual representations offer a full overview of the performance parameters, making the data more accessible and intuitive. Notably, these results show a persistent pattern in which the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model outperforms, especially when performed with a batch size of 0.4. Figure 6, which corresponds to Table 6, summarizes the model's performance across multiple criteria, highlighting the effect of varied batch sizes on its efficacy. Similarly, Figs. 7 and 8 provide visual interpretations of the results reported in Tables 7 and 8, respectively. The graphical representations are a valuable supplement to the numerical data, allowing for a more nuanced evaluation of the model's performance. A consistent pattern emerges across all three images, emphasizing the superiority of the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model when applied with a batch size of 0.4. This constancy in performance highlights the importance of batch size optimization, which appears to play a critical role in improving the model's overall effectiveness. These visual representations not only improve the interpretability of the findings, but also confirm

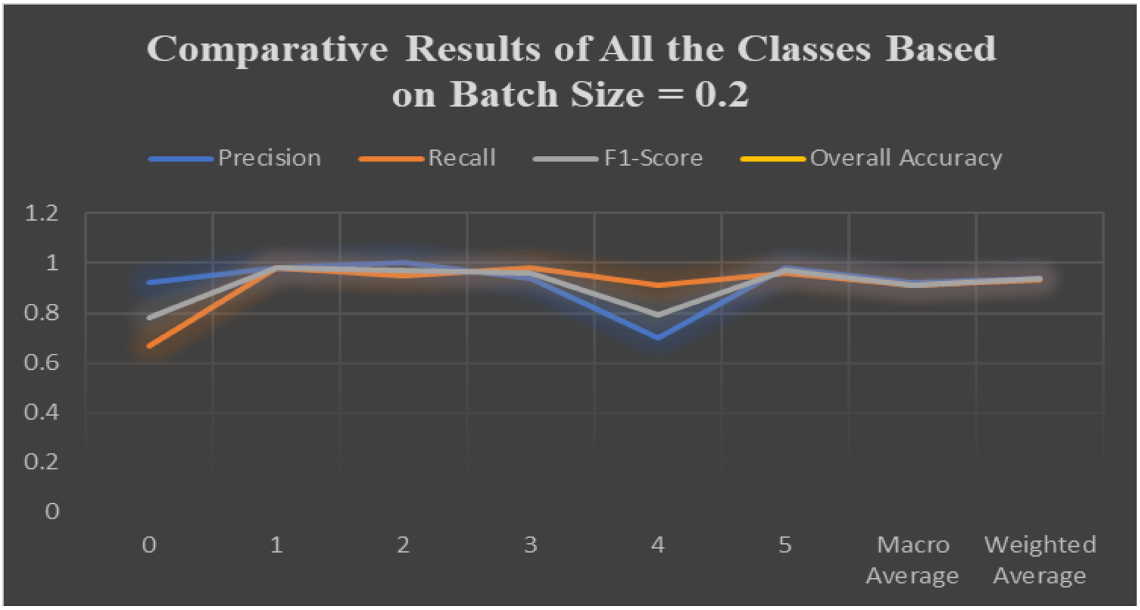
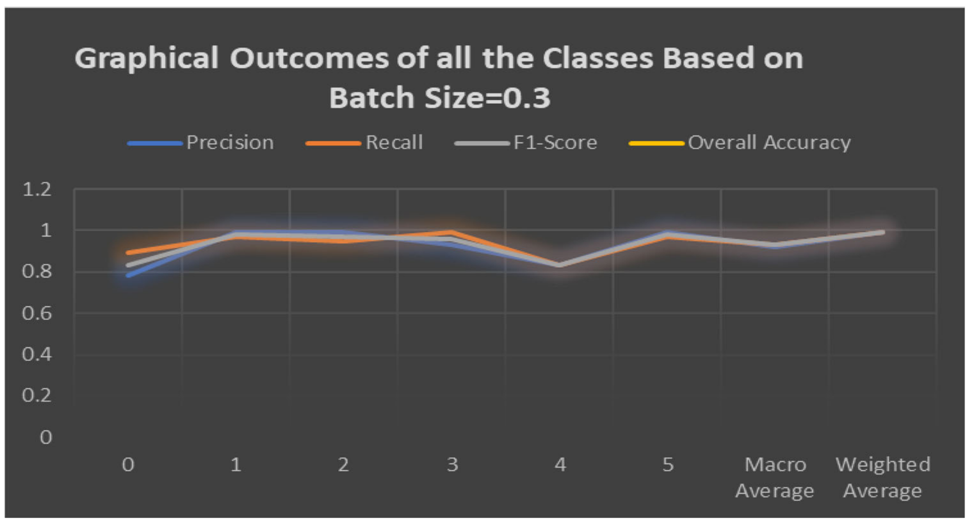


Fig. 6 Graphical representation of Table 6

Fig. 7 Graphical representation of Table 7



the conclusion that the CNN-BiLSTM-GRU model performs best when designed with a batch size of 0.4.

5.3.3 Computed Convergence Curves of Proposed Model Based on Confusion Matrix

We produced convergence curves for the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model in a systematic manner, observing its behavior over training iterations. These charts depicted the model's changing performance, revealing learning processes and potential convergence trends. During the training stages, monitoring indicators, such as loss functions and accuracy rates, allowed for a more detailed knowledge of the optimization process. These convergence curves were used as a diagnostic tool, revealing problems and leading revisions to improve the model's overall performance and robustness when dealing with complicated data patterns. The convergence curves for various batch sizes, shown in figures 9(a), 9(b) and 9(c), exhibit significant performance differences. Specifically, the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) hybrid deep learning architecture achieves its best performance with a batch size of 0.3 among the other batch sizes tested. When compared to other batch sizes, the curves associated with this batch size consistently display improved convergence properties. This finding emphasizes the importance of selecting an appropriately tuned batch size while optimizing the model's learning process. The use of a batch size of 0.3 not only improves con-

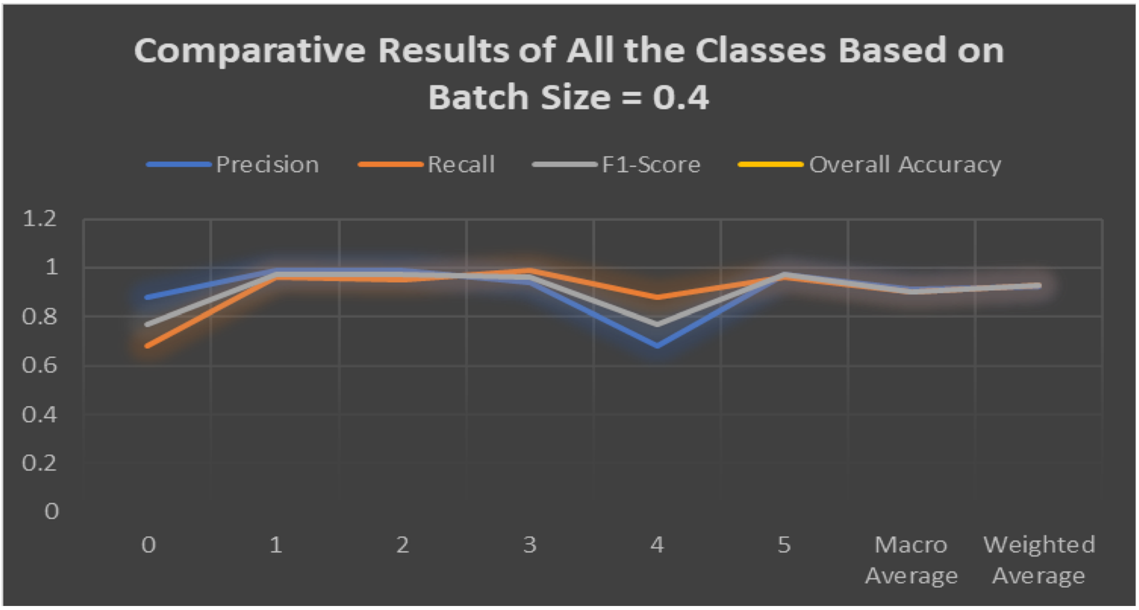
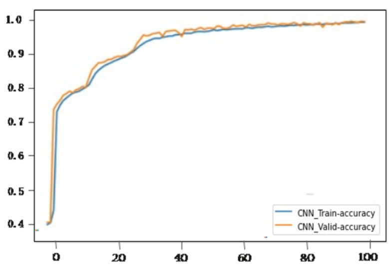
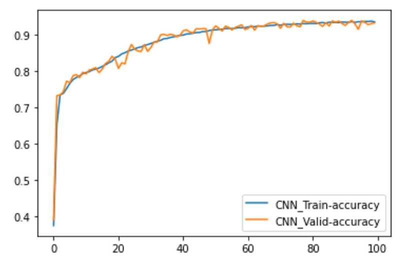


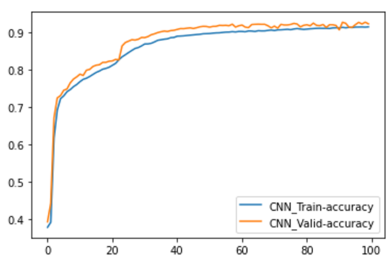
Fig. 8 Graphical representation of Table 8



(a) Convergence curves of All the Classes Based on Batch Size = 0.3



(b) Convergence curves of All the Classes Based on Batch Size = 0.2



(c) Convergence curves of All the Classes Based on Batch Size = 0.4

Fig. 9 Calculated convergence curves of based on confusion Matrix

vergence but also suggests a more efficient and effective use of the training data, which contributes to the CNN-BiLSTM-GRU model's overall performance in handling complicated tasks. The insights gained from these convergence curves in terms of batch size optimization provide useful direction for refining the model's training parameters and, as a result, enhancing its performance in real-world applications.

5.3.4 Compare our Proposed Model with the Statistical Analysis of the ANOVA test

In data analysis and validation, the ANOVA test and the CNN-BiLSTM-GRU hybrid deep learning model have different but complimentary functions. If there are any significant differences between the means of these groups, they can be found using ANOVA, a statistical method for comparing

Table 9 Comparative results of ANOVA test based on batch size = 0.2

Class	Precision	Recall	F1 Score	Overall accuracy
0	0.93	0.678	0.7845	0.9345
1	0.9768	0.9878	0.982	
2	0.99	0.95	0.967	
3	0.94	0.9845	0.974	
4	0.704	0.9123	0.7934	
5	0.982	0.962	0.978	
Macro-average	0.9165	0.9023	0.9023	
Weighted average	0.942	0.9237	0.9434	

Table 10 Comparative results of ANOVA test based on batch size = 0.3

Class	Precision	Recall	F1 Score	Overall accuracy
0	0.782	0.896	0.836	0.99784
1	0.9876	0.982	0.9856	
2	0.99	0.9445	0.9789	
3	0.9234	0.9923	0.9578	
4	0.85	0.8356	0.85	
5	0.9789	0.98	0.9896	
Macro-average	0.93	0.93	0.95	
Weighted average	0.9867	0.989	0.9934	

Table 11 Comparative results of ANOVA test based on batch size = 0.4

Class	Precision	Recall	F1 Score	Overall accuracy
0	0.882	0.70	0.80	0.9267
1	0.986	0.9567	0.9734	
2	0.992	0.9423	0.946	
3	0.957	0.9867	0.95	
4	0.70	0.8878	0.8023	
5	0.98	0.97	0.986	
Macro-average	0.9023	0.92	0.93	
Weighted average	0.926	0.94	0.926	

the means of several groups. When attempting to comprehend the impact of categorical variables on a continuous outcome, it is an effective method. The CNN-BiLSTM-GRU model, on the other hand, is made to handle complex, high-dimensional data, especially in applications like Human Activity Recognition (HAR). It combines Convolutional Neural Networks (CNN) for feature extraction, Bidirectional Long Short-Term Memory (BiLSTM) for capturing sequential dependencies, and Gated Recurrent Unit (GRU) for efficient sequence modeling. In a recent analysis, the CNN-BiLSTM-GRU model's performance was evaluated by using ANOVA alongside it. The results showed that the CNN-BiLSTM-GRU model performed similarly to the ANOVA test, which is remarkable because it suggests that the model's classifications and predictions are statistically trustworthy. The model's capacity to manage complex data patterns is demonstrated by this equivalency, which also attests to the model's performance meeting strict statistical criteria. Furthermore, the investigation showed that fine-tuning particular parameters had a significant impact on the CNN-BiLSTM-GRU model's performance. For example, utilizing a branch size of 0.3 in the model produced better results than branch sizes of 0.2 or 0.4. The model's accuracy and resilience were further enhanced by this fine-tuning, which elevated it to the status of an outstanding tool for jobs requiring the analysis of both temporal and spatial data. The CNN-BiLSTM-GRU model is one of the best-performing models for complicated data problems, and its performance was confirmed by ANOVA testing. Efficacy and dependability of the hybrid model are demonstrated by its capacity to meet the statistical rigor of ANOVA, especially in applications that need exact and accurate predictions, such Human Activity Recognition. The CNN-BiLSTM-GRU model is the best in data science and machine learning, as demonstrated by its combination of statistical validation and deep learning capabilities.Tables 9, 10, 11 provide a comparison of our suggested model and the statistical ANOVA test. In Figs. 10, 11, 12, the ANOVA test findings are further demonstrated by graphical interpretations.

5.4 Discussion

The performance of the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model across multiple batch sizes must be discussed in order to maximize its efficiency in handling difficult tasks. Batch size is important in the training dynamics of deep learning models, and experimenting with different numbers yields useful insights. Analyzing convergence curves, especially those shown in 9(a), 9(b), and 9(c), reveals significant patterns. Notably, the model performs well with a batch size of 0.3, emphasizing the need of carefully selecting batch sizes for enhanced convergence and overall efficacy. This topic focuses on how the ideal batch size influences the model's efficient learning from training data, determining its performance in addressing complex problems. The findings are critical for practitioners and researchers since they will guide them in fine-tuning the model's parameters for improved real-world performance.

5.4.1 Advantages of Our Proposed Methodology Compared to Current Approaches

The proposed methodology offers the following advantages over existing methods:

- Our hybrid deep learning strategy outperforms previous methods [25], [13], [14], [31], [30], [29], [36], in terms of classification accuracy.
- The previous technique advocated using a customized RF-federated architecture, evaluating three models, with the PP-FPRF demonstrating the most favorable performance as documented in [16]. In contrast, our newly presented Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model outperforms the prior PP-FPRF model. This represents a significant improvement in our model's capabilities, demonstrating greater efficacy and efficiency in dealing with the intricacies of the given assignment. The comparison emphasizes model development progress, emphasizing the CNN-BiLSTM-GRU model's potential to outperform earlier designs and contribute to field advancements.
- Before beginning the analysis, the data is preprocessed, which includes noise reduction, segmentation, and the use of a sliding window approach. The sliding window method, as described in [37], is used to divide sensor data into equal-sized windows.
- The Convolutional Neural Network-Bidirectional Long Short Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) architecture was used in the study to extract features from human behaviors. This methodology exhibits skill in capturing the intrinsic oscillations of certain activities while leveraging cross-domain knowledge. The model improves its ability to discover and express intricate patterns inside human behaviors by including Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory networks (BiLSTM), and Gated Recurrent Units (GRU). This technique, which makes use of cross-domain knowledge, provides a robust mechanism for characterizing and understanding the complex dynamics associated with numerous human activities.

5.4.2 Evaluating the Efficacy of our Proposed Methodology in Comparison to Previous Approaches

In this section, we use a sophisticated Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) hybrid deep learning architecture, which outperforms current deep learning and machine learning models. Table 12 demonstrates that our technique outperforms established methodologies, highlighting its originality.

6 Conclusion and Future Findings

Finally, our thorough research on the performance of the Convolutional Neural Network-Bidirectional Long Short-Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) model with varied batch sizes provides critical insights on optimizing deep learning architectures. By rigorously analyzing convergence curves, we discovered a persistent trend in which a batch size of 0.3 consistently outperformed other batch sizes, highlighting the critical role of parameter fine-tuning in improving model efficacy. This revelation highlights a key advance in the field, demonstrating the model's improved performance, particularly when working with the determined optimal batch size. These developments are critical in pushing the frontiers of deep learning approaches, allowing for more sophisticated and efficient neural network architectures. Looking ahead, our findings pave the way for additional research across a variety of fields. Future research could look into the subtle effects of batch size differences on a variety of activities and data qualities. This comprehensive methodology seeks to generalize our findings, assuring their robustness and usefulness across a diverse range of datasets and settings. Furthermore, our approach provides the framework for future research into dynamic batch size strategies, which dynamically modify

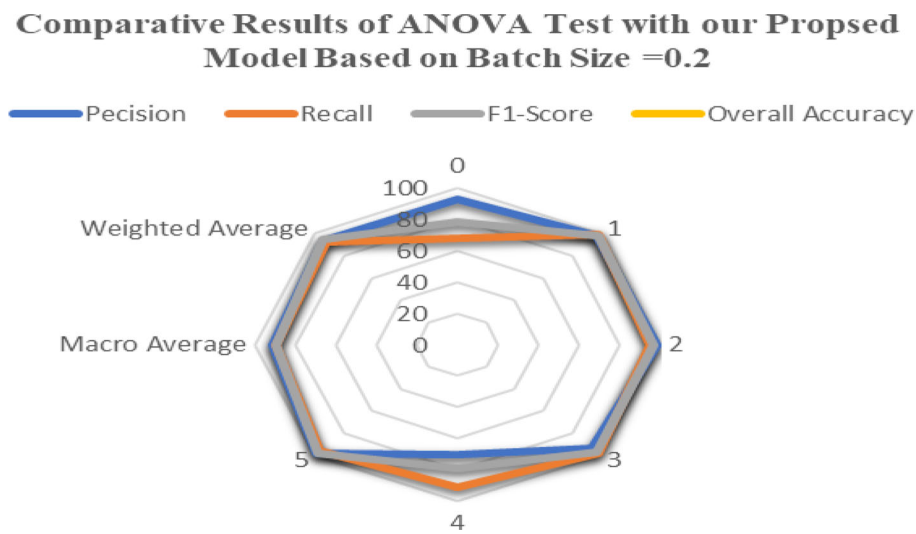


Fig. 10 Graphical representation of Table 9

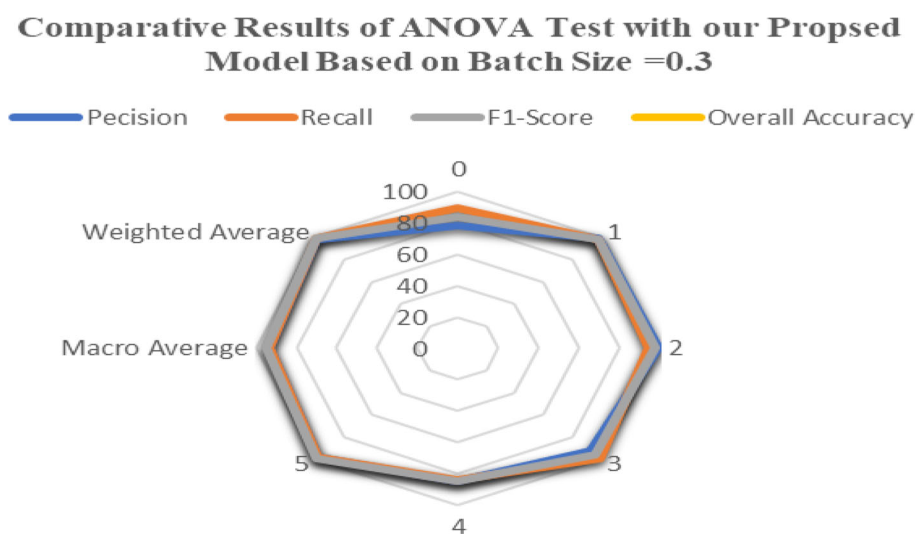


Fig. 11 Graphical representation of Table 10

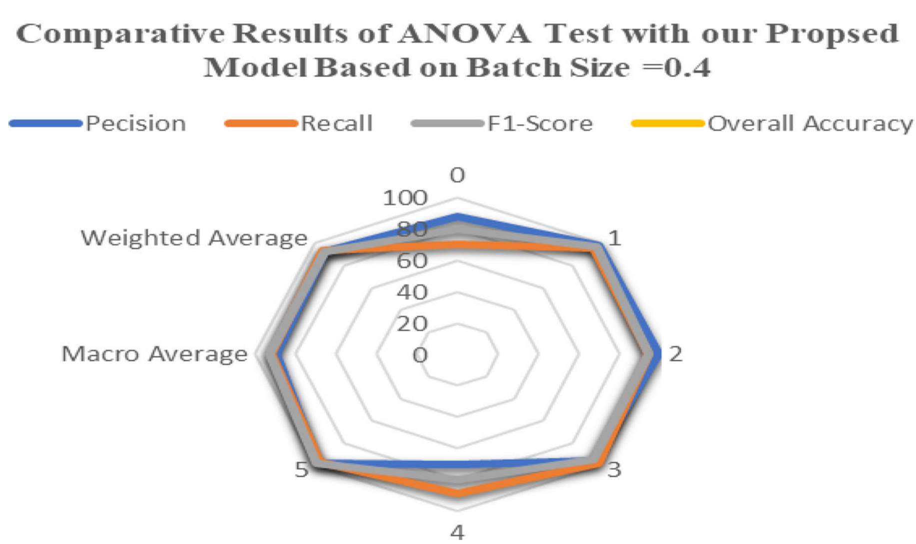


Fig. 12 Graphical representation of Table 11

Table 12 Analyzing and Contrasting Our Proposed Approach with Exciting Research

Author Name / References	Title	Dataset	Model	Accuracy
Songfeng, Liu et al. [16]	Federated Customised Random Forest for Human Behaviour Identification	Wisdm Dataset	PP-FPRF	94.5%
Z Zhongkai et al. [25]	A Comparative Study: Improving an Efficient Convolutional Neural Network Structure for Sensor-Based Human Activity Recognition	Wisdm Dataset	InceptionV3, DenseNet121	91.4%
Nayak Suvra et al. [26]	Comparative examination of the HAR dataset using classification algorithms	Wisdm Dataset	RF, SMO, SL	90.69%
Nidhi Dua et al. [14]	Using Inception-inspired CNN-GRU hybrid network to recognize human activity	Wisdm Dataset	CNN-GRU	97.21%
Kishor H Walse et al. [33]	An investigation into human activity recognition using AdaBoost classifiers on the Wisdm dataset	Wisdm Dataset	RF, REP Tree	94.61
Dipanwita Thakur et al. [31]	Human Activity Recognition Systems Based on Audio-Video Data Using machine learning and Deep learning	Wisdm dataset	ConvAE-LSTM	98.67%
Our proposed model	A Novel CNN-BiLSTM-GRU Hybrid deep learning model for Human activity recognition	Wisdm dataset	CNN-BiLSTM-GRU	**99.7%**

batch sizes during training to improve model flexibility and performance under changing situations.

Furthermore, introducing attention methods into our model design provides an intriguing opportunity to improve learning capabilities by allowing the model to focus on significant features and patterns in the data. Furthermore, investigating transfer learning algorithms shows promise in harnessing knowledge from pre-trained models, allowing for faster convergence and better generalization on new tasks. These proposed future initiatives not only contribute to our understanding of deep learning model optimization, but also have significant consequences for real-world applications. By constantly optimizing Convolutional Neural Network-Bidirectional Long Short Term Memory-Gated Recurrent Unit (CNN-BiLSTM-GRU) settings and embracing novel approaches, we are positioned to attain unparalleled levels of performance and efficiency in addressing complex data-driven challenges across multiple domains.

Funding Both authors indicate that this work received no specific funding from public, private, commercial, or non-profit organizations.

Data Availability https://www.kaggle.com/datasets/drsaeedmohsen/wisdmdataset2021?resource=download.

Declarations

Conflict of Interest There is no conflict of interest between the authors.

Ethical Approval The authors of this article did not conduct any studies with human volunteers or animals.

References

1. Abdel-Salam, R., Mostafa, R., Hadhood, M.: Human activity recognition using wearable sensors: review, challenges, evaluation benchmark. In: International Workshop on Deep Learning for Human Activity Recognition, pp. 1–15 (2021). Springer
2. Yao, L., Sheng, Q.Z., Benatallah, B., Dustdar, S., Wang, X., Shemshadi, A., Kanhere, S.S.: Wits: an iot-endowed computational framework for activity recognition in personalized smart homes. Computing **100**(4), 369–385 (2018)
3. Mousse, M.A., Motamed, C., Ezin, E.C.: Percentage of human-occupied areas for fall detection from two views. Vis. Comput. **33**(12), 1529–1540 (2017)
4. Mishra, A., Sharma, S., Kumar, S., Ranjan, P., Ujlayan, A.: Effect of hand grip actions on object recognition process: a machine learning-based approach for improved motor rehabilitation. Neural Comput. Appl. **33**(7), 2339–2350 (2021)
5. Vishwakarma, D.K., Dhiman, C.: A unified model for human activity recognition using spatial distribution of gradients and difference of gaussian kernel. Vis. Comput. **35**(11), 1595–1613 (2019)
6. Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human action recognition from various data modalities: A review. IEEE transactions on pattern analysis and machine intelligence (2022)
7. Chakraborty, P., Yousuf, M.A., Biswas, D., Azad, A., Moni, M.A.: Understanding the level of human attention for cognitive analysis to transplant the human behavioral mechanism to robots. Available at SSRN 4292648
8. Dewangan, D.K., Sahu, S.P.: Potnet: Pothole detection for autonomous vehicle system using convolutional neural network. Electron. Lett. **57**(2), 53–56 (2021)
9. Chen, L., Wang, R., Yang, J., Xue, L., Hu, M.: Multi-label image classification with recurrently learning semantic dependencies. Vis. Comput. **35**, 1361–1371 (2019)
10. Zhu, R., Tu, X., Huang, J.: Using deep learning based natural language processing techniques for clinical decision-making with ehrs. Deep learning techniques for biomedical and health informatics, 257–295 (2020)
11. Panwar, M., Dyuthi, S.R., Prakash, K.C., Biswas, D., Acharyya, A., Maharatna, K., Gautam, A., Naik, G.R.: Cnn based approach for activity recognition using a wrist-worn accelerometer. In: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2438–2441 (2017). IEEE
12. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
13. Bozkurt, F.: A comparative study on classifying human activities using classical machine and deep learning methods. Arab. J. Sci. Eng. **47**(2), 1507–1521 (2022)
14. Dua, N., Singh, S.N., Semwal, V.B.: Multi-input cnn-gru based human activity recognition using wearable sensors. Computing **103**(7), 1461–1478 (2021)
15. Guan, Y., Plötz, T.: Ensembles of deep lstm learners for activity recognition using wearables. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies **1**(2), 1–28 (2017)
16. Liu, S., Wang, J., Zhang, W.: Federated personalized random forest for human activity recognition. Math. Biosci. Eng **19**, 953–971 (2022)
17. Karim, F., Majumdar, S., Darabi, H., Chen, S.: Lstm fully convolutional networks for time series classification. IEEE Access **6**, 1662–1669 (2017)
18. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
19. Zhao, Y., Yang, R., Chevalier, G., Xu, X., Zhang, Z.: Deep residual bidir-lstm for human activity recognition using wearable sensors. Math. Prob. Eng. **2018**, 1–13 (2018)
20. Wan, S., Qi, L., Xu, X., Tong, C., Gu, Z.: Deep learning models for real-time human activity recognition with smartphones. Mobile Netw. Appl. **25**, 743–755 (2020)
21. Ignatov, A.: Real-time human activity recognition from accelerometer data using convolutional neural networks. Appl. Soft Comput. **62**, 915–922 (2018)
22. Wang, K., He, J., Zhang, L.: Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors. IEEE Sensors J. **19**(17), 7598–7604 (2019)
23. Teng, Q., Wang, K., Zhang, L., He, J.: The layer-wise training convolutional neural networks using local loss for sensor-based human activity recognition. IEEE Sensors J. **20**(13), 7265–7274 (2020)
24. Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J.: Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International Conference on Mobile Computing, Applications and Services, pp. 197–205 (2014). IEEE
25. Zhongkai, Z., Kobayashi, S., Kondo, K., Hasegawa, T., Koshino, M.: A comparative study: Toward an effective convolutional neural network architecture for sensor-based human activity recognition. IEEE Access **10**, 20547–20558 (2022)
26. Nayak, S., Panigrahi, C.R., Pati, B., Nanda, S., Hsieh, M.-Y.: Comparative analysis of har datasets using classification algorithms. Comput. Sci. Inform. Syst. **19**(1), 47–63 (2022)
27. Kaya, Y., Topuz, E.K.: Human activity recognition from multiple sensors data using deep cnns. Multimedia Tools Appl. **83**(4), 10815–10838 (2024)
28. Zhu, S., Chen, W., Liu, F., Zhang, X., Han, X., et al.: Human activity recognition based on a modified capsule network. Mobile Information Systems **2023** (2023)
29. Islam, M., Nooruddin, S., Karray, F., Muhammad, G., et al.: Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges and future prospects. arXiv preprint arXiv:2202.03274 (2022)
30. Tang, Y., Zhang, L., Teng, Q., Min, F., Song, A.: Triple cross-domain attention on human activity recognition using wearable sensors. IEEE Transactions on Emerging Topics in Computational Intelligence (2022)
31. Thakur, D., Biswas, S., Ho, E.S., Chattopadhyay, S.: Convae-lstm: Convolutional autoencoder long short-term memory network for smartphone-based human activity recognition. IEEE Access **10**, 4137–4156 (2022)
32. Helmi, A.M., Al-Qaness, M.A., Dahou, A., Damaševičius, R., Krilavičius, T., Elaziz, M.A.: A novel hybrid gradient-based optimizer and grey wolf optimizer feature selection method for human

activity recognition using smartphone sensors. Entropy **23**(8), 1065 (2021)
33. Walse, K.H., Dharaskar, R.V., Thakare, V.M.: A study of human activity recognition using adaboost classifiers on wisdm dataset. Inst Integr Omics Appl Biotechnol J **7**(2), 68–76 (2016)
34. Challa, S.K., Kumar, A., Semwal, V.B.: A multibranch cnn-bilstm model for human activity recognition using wearable sensor data. Vis. Comput. **38**(12), 4095–4109 (2022)
35. Onyekpe, U., Palade, V., Kanarachos, S., Christopoulos, S.-R.G.: A quaternion gated recurrent unit neural network for sensor fusion. Information **12**(3), 117 (2021)
36. Saeed, U., Kumar, K., Laghari, A.A., Khuhro, M.A., Islam, N., Shaikh, G.M., Hussain, F., Shaikh, A.A.: A comparative analysis of classification techniques for human activity recognition using wearable sensors and smart-phones. EAI Endorsed Trans Pervas Health Technol **8**(30), 1–1 (2022)
37. Antar, A.D., Ahmed, M., Ahad, M.A.R.: Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review. In: 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 134–139 (2019). IEEE