Human Activity Recognition

Amirhossein Afkhami
Department of Mechanical Engineering,
University of Alberta,
email: afkhamia@ualberta.ca
Contribution = 33%

Arash Salemi
Department of Mechanical Engineering,
University of Alberta,
email: salemi1@ualberta.ca
Contribution = 33%

Amirhossein Iranmehr
Department of Mechanical Engineering,
University of Alberta,
email: iranmehr@ualberta.ca
Contribution = 33%

This study focuses on using smartphone inertial sensor data for Human Activity Recognition (HAR) and classifying human activities using optimized machine learning models. It starts with explaining the dataset, its experiment, and features. The data was then pre-processed, involving the use of Principal Component Analysis (PCA) to reduce dataset dimensionality, followed by dataset splitting and normalization using z-score normalization. For classification, Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) were utilized, and their performances were compared. Additionally, the model was further optimized using the grid search method and genetic algorithm.

Keywords: Human Activity Recognition, Inertial Sensor, Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Genetic Algorithm, Grid Search Optimization

1 Introduction

Human Activity Recognition (HAR) is a technology that involves identifying and classifying various human activities based on sensor data collected from wearable devices or other sensors. In other words Human activity recognition is an ability to interpret human body gesture or motion via sensors and determine human activity [1]. Common human behaviors are often identified through human activities.

HAR has important applications in various fields, including healthcare [2], sports [3], and smart homes. By using HAR technology, healthcare providers can monitor the physical activity of patients and assess their recovery progress. Sports professionals can use HAR to track athletes' performance and prevent injuries. The scope of activity recognition varies from understanding the actions of an individual to recognizing activities involving multiple people [4]. Additionally, it's important to acknowledge that creating effective HAR solutions requires a thorough understanding of the concepts, limitations, and challenges involved. Human activities are described as a series of actions repeated over time within an environment[5].

In recent years, there has been an exploration into creating solutions that use computational technologies and methods to recognize human activities (HAR) [6,7]. The HAR issue is commonly addressed as a classification problem, aiming to identify the activity an individual is performing at a specific moment. Consequently, many HAR solutions have been developed using AI methods, using machine learning techniques. This includes both shallow and deep learning algorithms, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Deeply-Connected Networks (DFN)[8].

Smartphones are frequently used to create HAR solutions due to their widespread availability and the variety of sensors they contain. They fall under the category of wearable computing, and are a key component in mobile computing-based HAR systems[9]. Inertial sensors like accelerometers and gyroscopes are particularly useful in smartphones. They capture data on acceleration and the direction of the human body's movement, respectively. These sensors enable the gathering of varied information about the user, which is helpful in identifying individual physical activities[10].

Smartphone-based HAR solutions would follow the process that includes stages like data gathering, segmentation and fusion, feature extraction and selection, and the classification models using machine learning algorithms[11]. Recently, the field of HAR has shifted towards adopting new deep learning techniques, altering the traditional methods used for feature selection and dimension reduction. These changes pertain to how features are extracted[8].

Recognizing human activities is akin to a standard pattern recognition system, encompassing a series of steps from data gathering to classifying

Version 1.18, December 15, 2023

activities. This involves transforming raw sensor data into effective models for classifying human activities. For smartphones with inertial sensors, the Human Activity Recognition (HAR) methodology can be categorized into two types based on machine learning: shallow algorithms (like SVM, KNN, and decision trees) and deep algorithms (such as CNN, RNN, RBM, SAE, DFN, and DBM). The key distinction between these methods lies in the feature extraction process – whether the features are manually or automatically extracted [12] .

Our research delves into Human Activity Recognition (HAR) using smartphone sensor data, beginning with defining specific activities to recognize. We rigorously explore various datasets, focusing on accelerometer and gyroscope data, crucial for identifying human movements. Clear research objectives guide our approach, emphasizing accuracy in activity recognition. The data processing phase involves cleaning and normalizing raw data for analysis, addressing challenges like noise reduction. We apply advanced machine learning algorithms, especially Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs), due to their effectiveness in sequential data analysis and capturing temporal activity patterns. Hyperparameter optimization is a key step, fine-tuning aspects like learning rates to enhance model performance.

2 Problem Definition

2.1 About Dataset. The Human Activity Recognition data we used was built from the recordings of 30 participants performing activities of daily living while carrying a waist-mounted smartphone with embedded inertial sensors. [13] The goal is to classify activities into one of the six activities performed.

2.2 Description of experiment. The dataset we used for HAR consisted of 30 participants ranging in age from 19 to 48 years old. Each individual performed six different activities as follows:

- Walking
- Standing
- Walking Downstairs
- Sitting
- Walking Upstairs
- Laying

The participants carried out these activities while wearing a **Samsung Galaxy S II** smartphone on their **waist**. The smartphone's built-in accelerometer and gyroscope were used to capture the 3-axial linear acceleration and 3-axial angular velocity at a consistent rate of 50Hz. Subjects experimented twice: once with the smartphone fixed on the left side of the belt, and once with the user placing it as preferred. There was a 5-second separation between each task for rest. The experiments were recorded to manually label the data. The dataset obtained from the experiments was randomly divided into two sets, with 70% of the volunteers selected for generating the training data and 30% for the test data. [13] Figure 1 shows the number of samples for each activity.

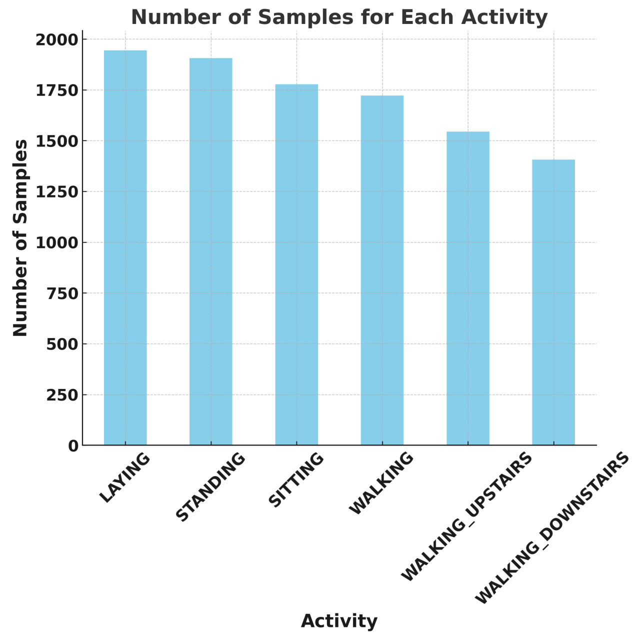


Fig. 1 Human activity classes and sample counts

2.3 Signal Processing. In the mentioned study [13], the sensor signals were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low-frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain (Table 1). [13]

The reason to sample data in fixed-width sliding windows of 2.56 sec is to accommodate the cadence of an average person walking, which falls within the range of 90 to 130 steps per minute. Each window sample aimed to capture a full walking cycle (two steps). [13].

Table 1 Time and frequency domain signals obtained from the smartphone sensors.

Name	Time	Freq
Body Acc	1	1
Gravity Acc	1	0
Body Acc Jerk	1	1
Body Angular Speed	1	1
Body Angular Acc	1	0
Body Acc Mag	1	1
Gravity Acc Mag	1	0
Body Acc Jerk Mag	1	1
Body Angular Speed Mag	1	1
Body Angular Acc Mag	1	1

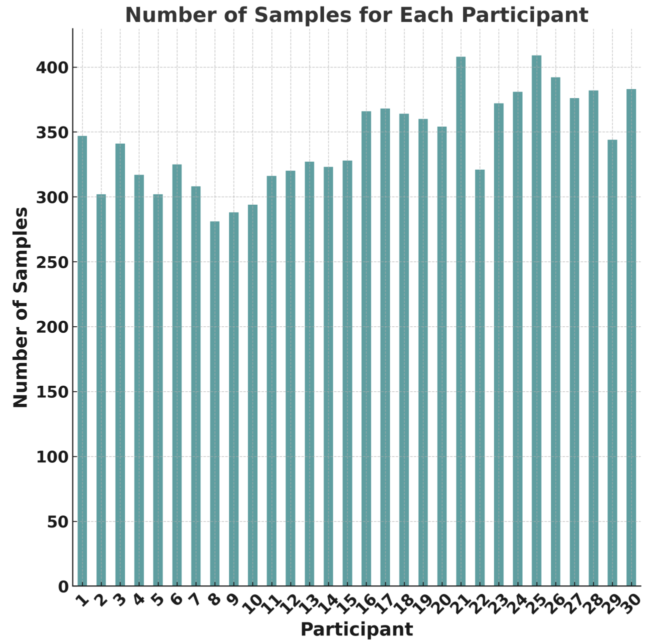
2.4 Dataset Features. In that study [13], a vector of features was obtained from each sampled window mentioned earlier. Commonly used measures in HAR literature such as the mean, correlation, signal magnitude area (SMA), and autoregression coefficients were employed for the feature mapping.[13] Table 2 provides a list of all measures applied to the time and frequency domain signals. In total, 561 features were extracted to describe each activity window.

Table 2 List of measures in feature vectors

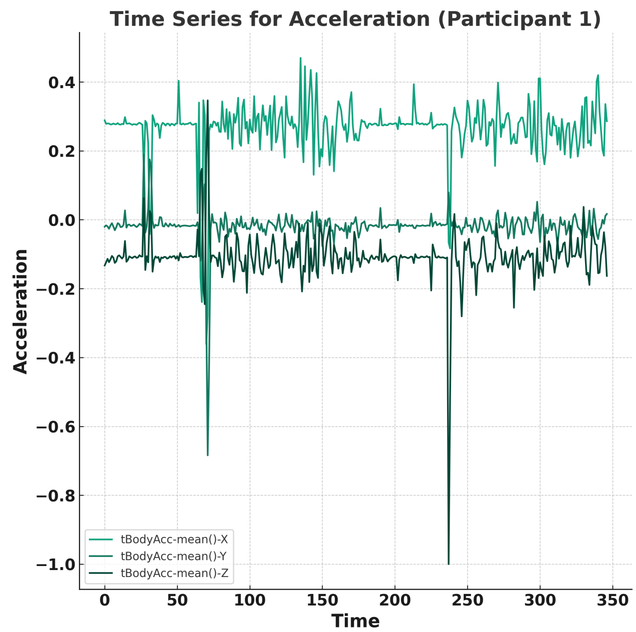
Function	Description
mean	Mean value
std	Standard deviation
mad	Median absolute value
max	Largest values in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of the squares
iqr	Interquartile range
entropy	Signal Entropy
arCoeff	Autorregresion coefficients
correlation	Correlation coefficient
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
skewness	Frequency signal Skewness
kurtosis	Frequency signal Kurtosis
energyBand	Energy of a frequency interval
angle	Angle between two vectors

To summarize, the dataset includes triaxial acceleration and angular velocity, a 561-feature vector, activity labels, and unique identifiers for the individuals conducting the experiment.

Figure 2a displays the number of samples collected from each participant, showing a consistent range of 270 to 410 samples. In Figure 2b, the mean body acceleration in the X, Y, and Z directions is depicted for participant one when performing various tasks.



(*a*)



(*b*)

Fig. 2 a) Training samples for all participants, b) Mean of Body Acceleration for participant 1 when performing different activities

2.5 Objectives. The primary objective of this project is to effectively classify human activities using advanced machine learning techniques. To achieve this, we will first implement data preprocessing methods, including normalization and Principal Component Analysis (PCA), to refine and streamline the dataset for optimal model performance. Our approach then involves the application of sophisticated Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs) to accurately classify various human activities captured in the dataset. An integral part of our methodology will be the optimization of our models using the grid search technique, ensuring that we achieve the highest possible accuracy and efficiency in our classification. This comprehensive strategy aims to demonstrate the efficacy of machine learning in the nuanced field of human activity recognition.

3 Methodology

In this section, we want to first pre-process our dataset using Principal Component Analysis (PCA) to reduce dataset dimensionality, followed by dataset splitting and normalization using the z-score normalization technique. Subsequently, we will utilize machine learning algorithms to classify human activities based on our dataset. Finally, we will optimize our DNN model's hyperparameters using the grid search method and genetic algorithm to improve its performance. The project framework is depicted in Figure 3.

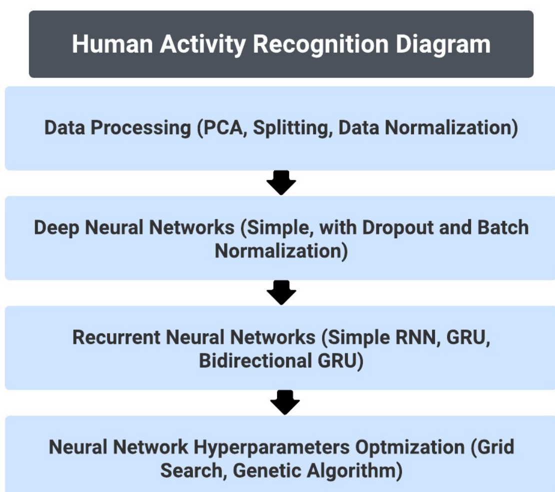


Fig. 3 Project framework

3.1 Data Processing. The objective of this section is to process the data captured by the inertial sensor of the smartphone and prepare it for machine learning algorithms. Initially, we will utilize Principle Component Analysis (PCA) to reduce the dimensionality of the dataset. Following that, we will split the dataset and normalize it using the z-score normalization technique.

3.1.1 Dimension Reduction. In our study, we are examining a dataset with 561 features, which suggests potential correlations between them. To understand these correlations, we began by creating a visual representation of the correlation matrix, focusing on a selected subset of features for simplicity. This matrix, illustrated in Figure 3a, displays correlation coefficients between pairs of features. This approach helps in identifying patterns and potential redundancies in the dataset. Understanding these correlations is crucial for our next steps, including applying dimensionality reduction techniques like Principal Component Analysis (PCA) and informing our strategy for feature selection and model optimization in our machine learning algorithms, specifically Deep Neural Networks (DNNs) and Recurrent Neural Networks (RNNs).

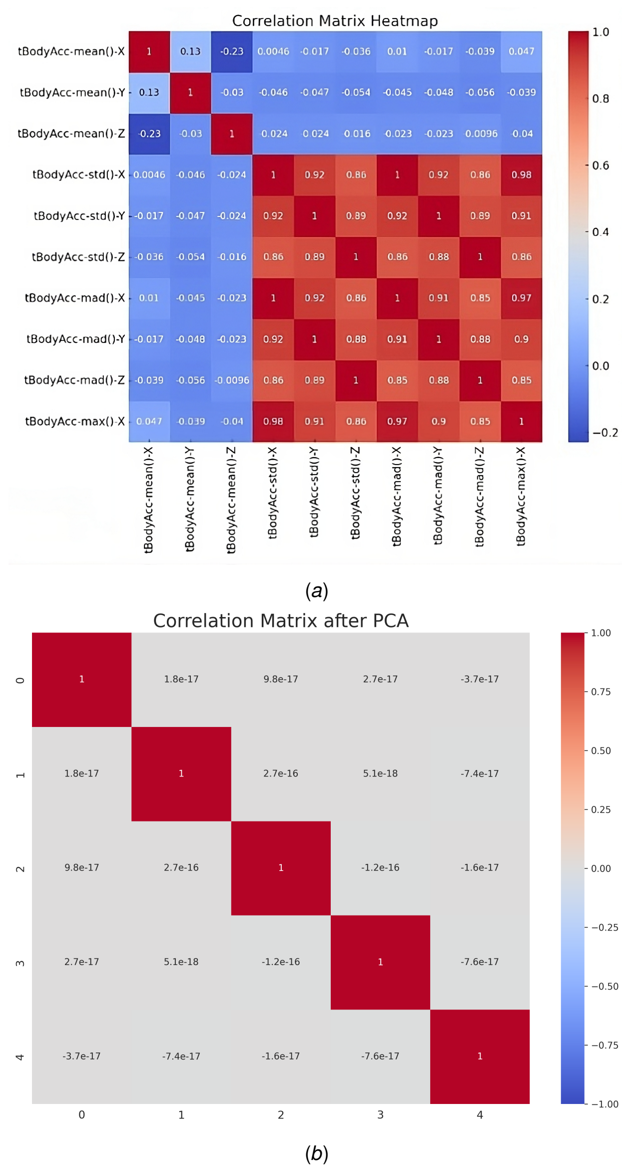


Fig. 4 (a) Original correlation matrix, and (b) the correlation matrix post PCA.

Based on Figure 4a, it is evident that there is a strong correlation among the features. To address this issue, we utilized Principle Component Analysis (PCA) to reduce the dimensionality of the dataset. Figure 5 displays the results of PCA analysis, while Figure 4b depicts the confusion matrix for the best five PCA components. These figures indicate that the top five PCA components have very low correlation and can retain a high amount of the dataset information.

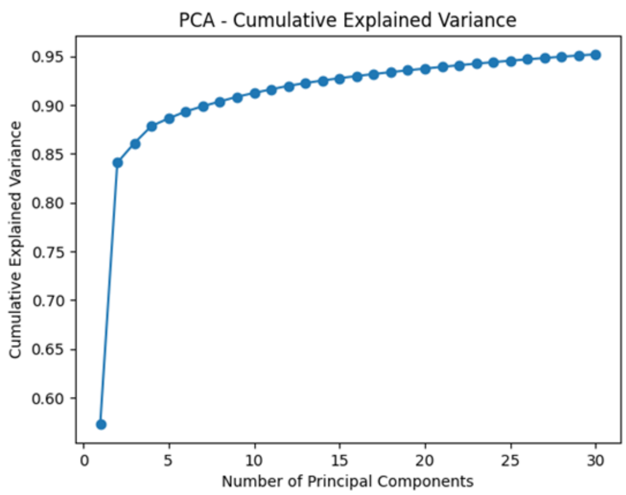


Fig. 5 PCA components cumulative information retained

According to Figure 5, the top five components can preserve almost 90% of the information in the entire dataset. Based on this observation, we decided to select only the 5 best components and proceed with the project using them.

3.1.2 Data Splitting. After normalizing the dataset, we divided it into training, validation, and testing sets. The proportions were 70% for training, and 15% each for testing and validation. This split offers benefits such as providing sufficient training data, unbiased evaluation of model performance, and fine-tuning of hyperparameters without bias. It also helps prevent overfitting and ensures the model's ability to generalize to new data.

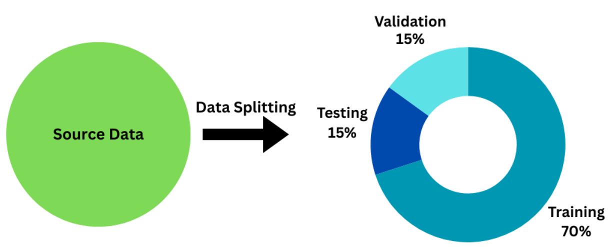


Fig. 6 Splitting dataset into training, validation, and testing sets

3.1.3 Normalization. After partitioning the dataset into training, validation, and testing sets, the crucial step of data normalization is performed. We chose Z-score normalization (also known as standardization) for this purpose. This technique standardizes the data to ensure that each feature has a mean of 0 and a standard deviation of 1. The formula for Z-score normalization is expressed as:

$$\hat{x} = \frac{x - \mu}{\sigma} \tag{1}$$

Here, $\hat{x}$ represents the normalized value, x is the original value, μ is the mean, and σ is the standard deviation of the dataset. This process is crucial for aligning the scales of different features, particularly in a dataset with varying ranges of sensor readings.

3.2 Machine Learning Algorithms.

In this section, our objective is to employ machine learning algorithms, particularly Neural Networks (NNs), to classify human activities using our dataset. Initially, we developed Deep Neural Networks (DNN) and incorporated techniques such as dropout and batch normalization to improve the network's performance. Additionally, we further enhanced the network's performance by using different types of Recurrent Neural Networks (RNNs).

3.2.1 Deep Neural Networks. Initially, we developed a DNN architecture to classify human activities. The schematic of the architecture can be found in the Appendix. The architecture of Deep Neural Networks depicted in Appendix. Our training approach utilized the following hyperparameters, as outlined in the Table 3.

Table 3 The hyperparameters used for training the designed Deep Neural Networks (DNN)

Parameter	Value	Parameter	Value
Optimizer	Adam	Batch Size	128
Activation	ReLU	Early Stop	Patience 25
Loss	Cross Entropy	Validation	K-fold (k=3)
Learning Rate	0.001	# neurons	layer 1 = 4
Num Layers	2	# neurons	layer 2 = 3

Another component in our training is the use of K-fold validation. This technique enhances the robustness and reliability of model evaluation by ensuring that it is tested across different subsets of the data. In K-fold validation, the training data is divided into 'K' number of folds or subsets. The model is then trained K times using a different fold as the validation set and the remaining K-1 folds as the training set. The use of K-fold validation, in combination with other hyperparameters makes our DNN model learn effectively and generalize the patterns inherent in human activity recognition.

However, the results were not satisfying and the accuracy was about 80% (Figure 8a). That is why we added the following techniques to enhance the network performance:

- Dropout: It is a regularization technique to prevent overfitting. It randomly drops out some neurons in a layer during training with a certain probability (typically 0.5).
- Batch Normalization: It normalizes the activations of each layer in a network to have zero mean and unit variance. It can improve generalization by reducing the dependence of each layer on the distribution of inputs from the previous layer.

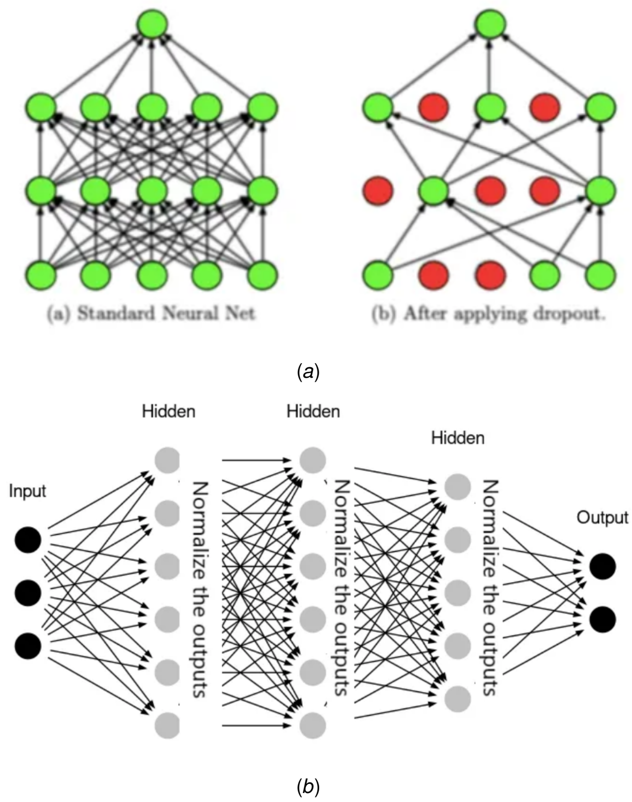


Fig. 7 (a) Dropout and (b) Batch Normalization techniques in neural networks.

After adding these methods, we trained the network again with the same parameters as shown in Table 3. The results were enhanced by 10% (Figure 8a).

However, based on the observed confusion matrix in Figure 8b, it is apparent that the network showed difficulty when classifying two specific human activities: standing and sitting. This phenomenon is due to the fact that in HAR, it's crucial to consider not just the current sensor readings, but also how these readings change and relate to each other over a period of time. Without this temporal understanding, a DNN might be unable to differentiate between activities that have similar individual readings but differ in their temporal patterns. That is why to further enhance the network's performance, we used Recurrent Neural Networks (RNNs) in the subsequent part.

3.2.2 Recurrent Neural Networks. Recurrent Neural Networks (RNNs) are a type of neural network that are designed to process sequential data. Unlike traditional feedforward neural networks, RNNs have feedback connections that allow them to store and utilize information from previous steps in the sequence. This makes RNNs particularly a good choice for this project since they can capture the sequence in the dataset. Specifically, we employed the following types of RNNs for classification in this project:

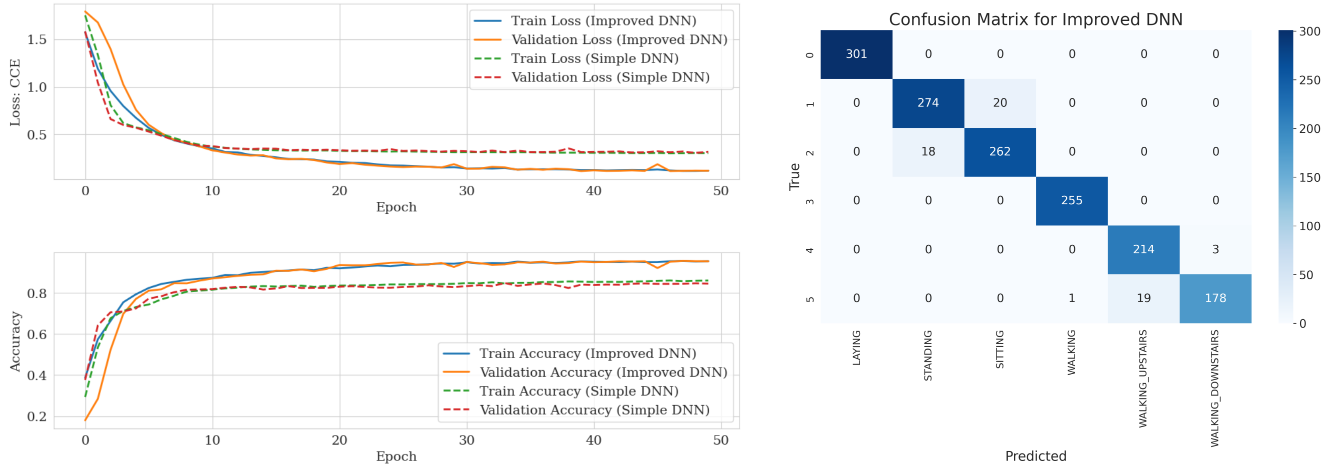
- Simple RNN
- GRU
- Bidirectional GRU

The Gated Recurrent Unit (GRU) is a specialized form of Recurrent Neural Network (RNN) tailored for sequence prediction tasks, making it well-suited for Human Activity Recognition (HAR). In HAR, GRU's architecture, with its efficient gating mechanisms, addresses the vanishing gradient problem typical in standard RNNs. It includes an input layer designed for sequential time-series data, a GRU layer with a specific number of hidden units to capture temporal dependencies.

Table 4 The hyperparameters used for training the designed Recurrent Neural Networks (RNNs)

Parameter	Value	Parameter	Value
Optimizer	Adam	Batch Size	128
Activation	ReLU	Early Stop	Patience 25
Loss	Cross Entropy	Validation	K-fold (k=3)
Learning Rate	0.001		

The Bidirectional GRU extends this concept by processing the input data in both forward and reverse directions, incorporating two GRU layers. This dual-layer approach allows the network to capture both past and future context, significantly enhancing the understanding of sequential patterns in HAR. The outputs of the two GRU layers are combined before classification, and the model is similarly trained with careful monitoring of loss and accuracy to avoid overfitting. Figure 9 is the schematic of the Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), and Bidirectional Gated Recurrent Unit (GRU).



(a) (b)

Fig. 8 (a) Loss and Accuracy for training and validation sets for DNN with and without batch normalization and dropout, (b) Confusion Matrix for Improved DNN

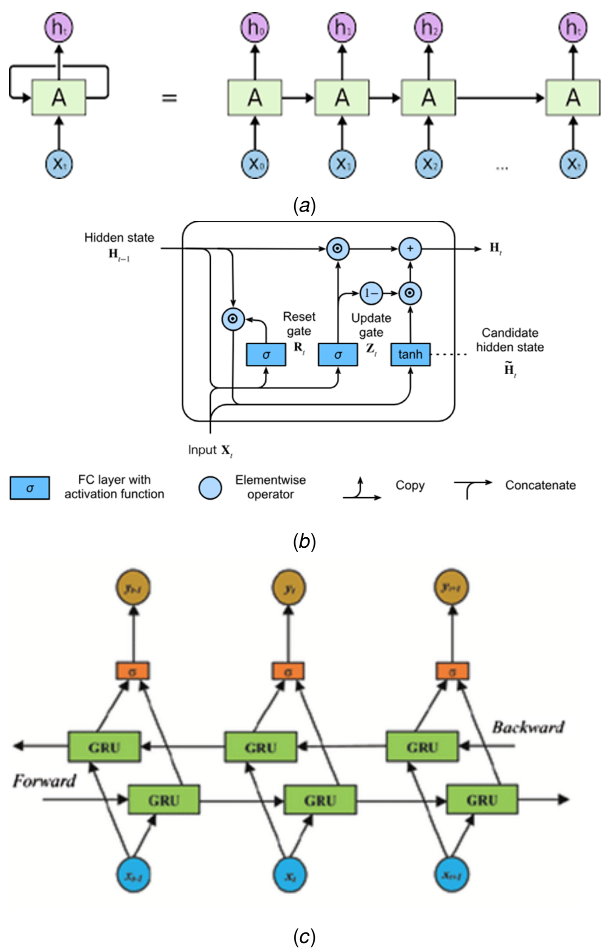


Fig. 9 Schematics of (a) RNN (b) GRU (c) Bidirectional Gated Recurrent Unit (GRU)

The RNN model initiates with a SimpleRNN layer, featuring 12 neurons and a rectified linear unit (ReLU) activation function. Engineered for sequences, this layer is configured to process input sequences with a shape of (3, 5), signifying sequences of a length of 3 and 5 features per time step. Notably, the configuration dictates that solely the output from the final time step is taken into consideration. Following the RNN layer, a Dense layer is introduced, comprising 6 units and a softmax activation function—ideal for multi-class classification tasks. The model is then prepared for training by compiling it with the Adam optimizer, set at a learning rate of 0.001, categorical crossentropy as the chosen loss function, and accuracy as the designated evaluation metric. The hyperparaemetr utilized for RNNs is illustrated in Table 4.

The results of loss and accuracy of training and validation for GRU and Bidirectional GRU is depicted in Figure 10 and Table 5. RNN (Recurrent Neural Network), GRU (Gated Recurrent Unit), and Bidirectional GRU. Both RNN and GRU have an identical F1 Score of 0.89, indicating comparable performance in precision and recall. However, the Bidirectional GRU surpasses them with a higher F1 Score of 0.92, likely due to its ability to process data in both directions, thus capturing more contextual information. In terms of accuracy, the GRU leads at 95%, demonstrating its effectiveness in classification tasks, followed by the Bidirectional GRU at 93%, which, despite its complexity, maintains high accuracy. The RNN, while still performing well, shows the lowest accuracy at 90%.

Table 5 Comparison of F1 and Accuray of RNN, GRU and Bidirectional GRU

Model	RNN	GRU	Bidirectional GRU
F1	0.89	0.89	0.92
Accuracy	90%	95%	93%

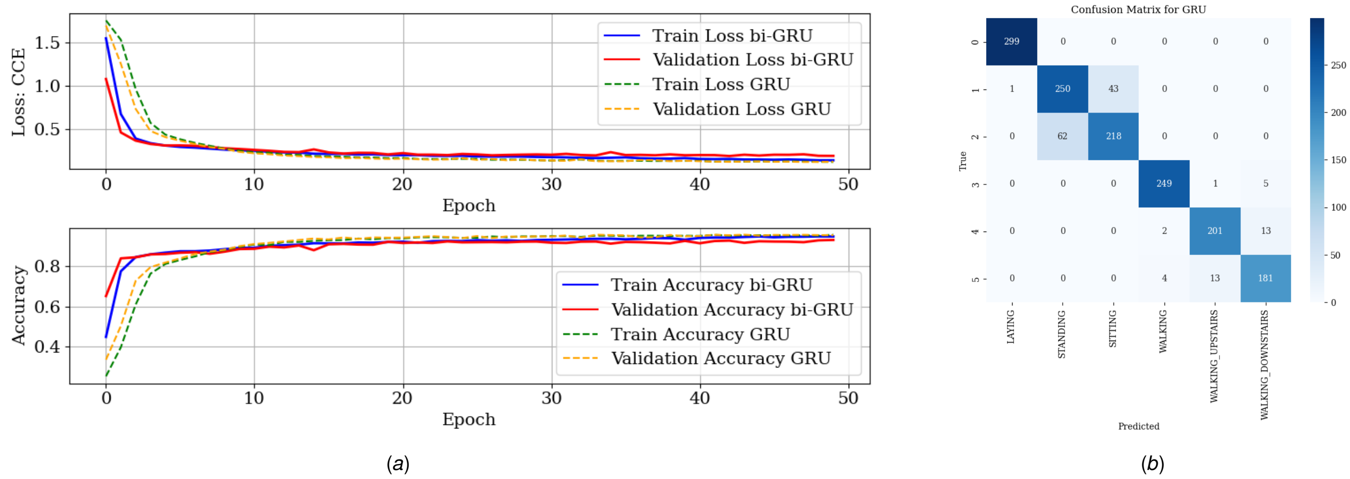


Fig. 10 (a) Loss and Accuracy for training and validation sets for GRU and Bidirectional GRU, (b) Confusion Matrix for the best Recurrent Neural Network Model (GRU)

4 Optimization

In our study, we employed grid search and genetic algorithm to systematically tune the hyperparameters of the Deep Neural Network. This process involved defining a grid of hyperparameter values and training models for each combination in this grid. We meticulously evaluated the performance of each model configuration using a cross-validation approach, ensuring that our selection of hyperparameters, such as the number of hidden layers, the number of neurons in each layer, learning rate, and dropout rate, was optimized for the best performance. This exhaustive search allowed us to identify the most effective hyperparameters for our DNN, leading to a more accurate and reliable human activity recognition model.

The tuning of the DNN's hyperparameters was a crucial step in enhancing the model's accuracy. We utilized a grid search technique and genetic algorithm, which entailed exploring a range of values for each hyperparameter. Key hyperparameters included the learning rate, the number of recurrent layers, the number of units in each layer, and the dropout rate. By training the DNN with various combinations of these parameters and assessing their impact on model performance, we were able to pinpoint the optimal configuration. The grid search approach ensured a comprehensive evaluation, as each parameter combination was rigorously tested under the same conditions, providing us with a robust and well-tuned model for classifying human activities. It should be note that we utilized **"Scikit learn GridSearchCV"**library for this study.

The genetic algorithm implemented in the provided code utilizes a fitness function based on the accuracy of a neural network, with the objective of optimizing the number of neurons in a hidden layer within the range of [5, 128]. The population size is set to 10 individuals, and the algorithm runs for 5 generations. A mutation rate of 0.2 and a crossover rate of 0.7 are employed, controlling the probabilities of mutation during crossover and mating, respectively. The fitness function evaluates the performance of a neural network with a specific number of neurons, and the genetic algorithm explores the solution space through the combination of blend crossover and Gaussian mutation. Tournament selection is used to choose individuals for reproduction based on their fitness values. The best individual, representing the optimal number of neurons, is extracted from the final population.

The optimized hyperparameters are shown in Table 5 as a result of our optimization process. It is important to note that all hyperparameters, except for the number of neurons, were optimized using "Grid search." In addition, the number of neurons was optimized using "Genetic Algorithm." The results for this optimization are illustrated in Table 5 and 6. Also, the training and validation results for DNN before and after optimization is shown in Figure 11.

Table 6 Neural Networks hyperparameters and their optimization methods

Parameter	Value	Optimization Method
Learning rate	0.1	Grid Search
Num neurons	79	Genetic Algorithm
Activation	eLu	Grid Search
Optimizer	Adam	Grid Search
Dropout	0	Grid Search
Num hidden layers	1	Grid Search

5 Results and Discussion

Our study has focused on the use of smartphone inertial sensor data for Human Activity Recognition (HAR) and has emphasized the classification of human

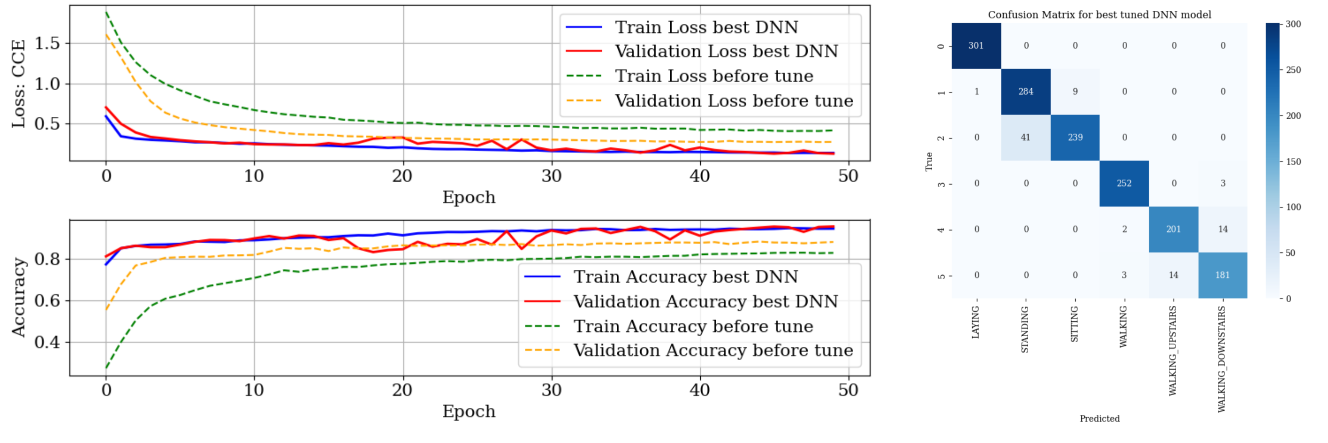


Fig. 11 **(a) Loss and Accuracy for training and validation sets for DNN before and after Optimization, (b) Confusion Matrix for Optimized DNN**

Table 7 **Comparison of Various Model Performances**

Model	Accuracy	F1 Score
Simple DNN	0.86	0.85
DNN	0.92	0.96
RNN	0.90	0.89
GRU	0.95	0.89
Bidirectional GRU	0.93	0.92
Tuned DNN	0.95	0.95

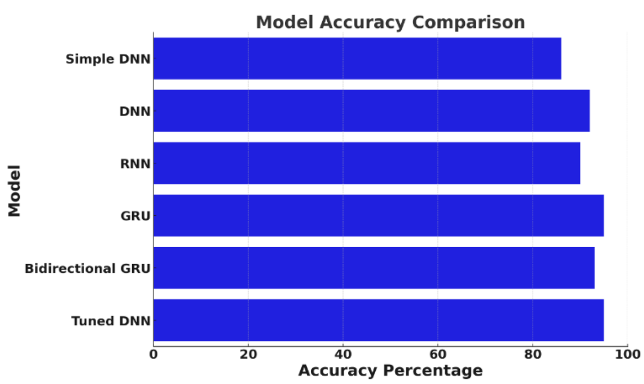


Fig. 12 **Comparison of different model performance**

activities using optimized machine learning models. We started by detailing the dataset and its experiment, followed by data preprocessing, which included Principal Component Analysis (PCA) for dimensionality reduction and z-score normalization. The classification was carried out using Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), and other variations like GRU and Bidirectional GRU. The optimization of these models was meticulously conducted using grid search and genetic algorithms.

From the obtained results shown in Table 6 and Figure 12 , we observe that the hyperparameters, including learning rate, number of neurons, and activation functions, were finely tuned using grid search and genetic algorithms. This optimization significantly enhanced the performance of the models. For instance, the DNN model achieved an accuracy of 92% and an F1 score of 0.96, while the GRU model showed an impressive accuracy of 95% with an F1 score of 0.89. The Bidirectional GRU also demonstrated robust performance with an accuracy of 93% and an F1 score of 0.92. Notably, the tuned DNN model, which integrated these optimized parameters, reached an accuracy and F1 score of 95% and 0.95 respectively, underscoring the effectiveness of the optimization process.

In conclusion, our study presents a significant advancement in HAR using smartphone sensors, showcasing how optimized machine learning models can substantially improve activity classification. The careful selection and tuning of hyperparameters have played a pivotal role in enhancing model performance. These findings not only contribute to the field of human activity monitoring but also set a strong foundation for future exploration and application in real-world scenarios. The methodologies and results detailed in this study highlight the potential of machine learning in transforming the landscape of activity recognition and monitoring using everyday technology. Finally, the concluding remarks of the project would be as follows:

(1) The project achieved high accuracy in activity recognition, notably with some models reaching 95% accuracy, marking a significant advancement in HAR using smartphone sensors.
(2) Hyperparameter tuning, particularly through grid search and genetic algorithms, was pivotal in improving model performance, underscoring its im-

portance in machine learning.

(3) Comparative analysis of various machine learning models provided insights for selecting the most effective model for different activity recognition scenarios.

(4) The successful classification of human activities opens opportunities for practical applications and sets a foundation for future research in enhancing HAR systems.

References

[1] Yang, J., Lee, J., and Choi, J., 2011, "Activity recognition based on RFID object usage for smart mobile devices," Journal of Computer Science and Technology, **26**(2), pp. 239–246.

[2] Fontecha, J., Navarro, F. J., Hervás, R., and Bravo, J., 2013, "Elderly frailty detection by using accelerometer-enabled smartphones and clinical information records," Personal and ubiquitous computing, **17**, pp. 1073–1083.

[3] Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J., 2015, "Activitynet: A large-scale video benchmark for human activity understanding," *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970.

[4] Ziaeefard, M. and Bergevin, R., 2015, "Semantic human activity recognition: A literature review," Pattern Recognition, **48**(8), pp. 2329–2345.

[5] Cook, D. J. and Krishnan, N. C., 2015, *Activity learning: discovering, recognizing, and predicting human behavior from sensor data*, John Wiley & Sons.

[6] Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., and Yu, Z., 2012, "Sensor-based activity recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), **42**(6), pp. 790–808.

[7] Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. T., 2010, "A survey of mobile phone sensing," IEEE Communications magazine, **48**(9), pp. 140–150.

[8] Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L., 2019, "Deep learning for sensor-based activity recognition: A survey," Pattern recognition letters, **119**, pp. 3–11.

[9] Bulling, A., Blanke, U., and Schiele, B., 2014, "A tutorial on human activity recognition using body-worn inertial sensors," ACM Computing Surveys (CSUR), **46**(3), pp. 1–33.

[10] Yang, J., 2009, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–10.

[11] Incel, O. D., Kose, M., and Ersoy, C., 2013, "A review and taxonomy of activity recognition on mobile phones," BioNanoScience, **3**(2), pp. 145–171.

[12] Sousa Lima, W., Souto, E., El-Khatib, K., Jalali, R., and Gama, J., 2019, "Human activity recognition using inertial sensors in a smartphone: An overview," Sensors, **19**(14), p. 3213.

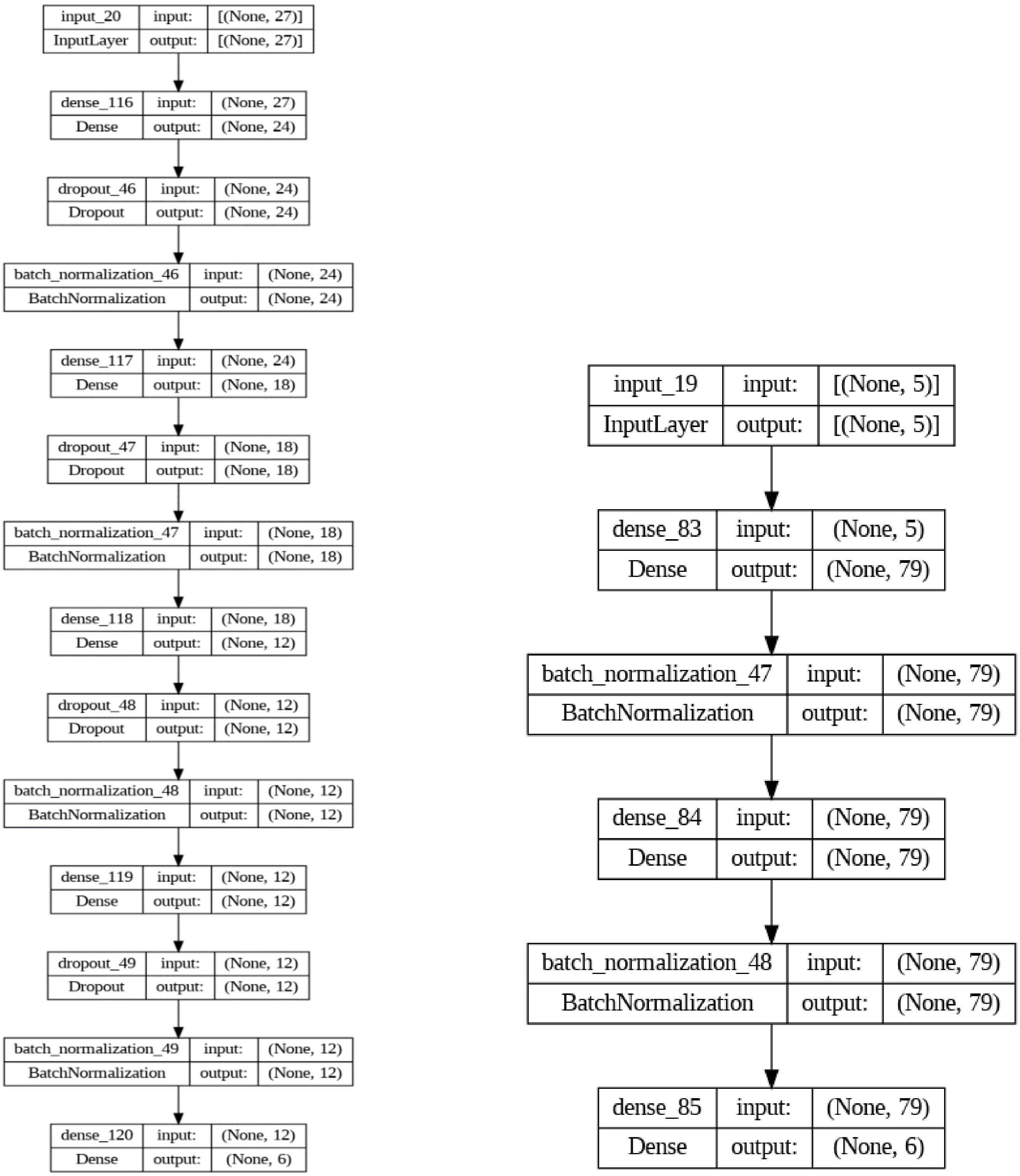
[13] Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes-Ortiz, J. L., et al., 2013, "A public domain dataset for human activity recognition using smartphones." *Esann*, Vol. 3, p. 3.

[14] Khan, W. Z., Xiang, Y., Aalsalem, M. Y., and Arshad, Q., 2012, "Mobile phone sensing systems: A survey," IEEE Communications Surveys & Tutorials, **15**(1), pp. 402–427.

[15] Vrigkas, M., Nikou, C., and Kakadiaris, I. A., 2015, "A review of human activity recognition methods," Frontiers in Robotics and AI, **2**, p. 28.

6 Appendix

The next pages display the architectures of DNN, Optimized DNN, RNN, GRU, and Bidirectional GRU models.



(*a*) (*b*)

Fig. 13 (a) DNN architecture, (b) Optimized DNN Architecture

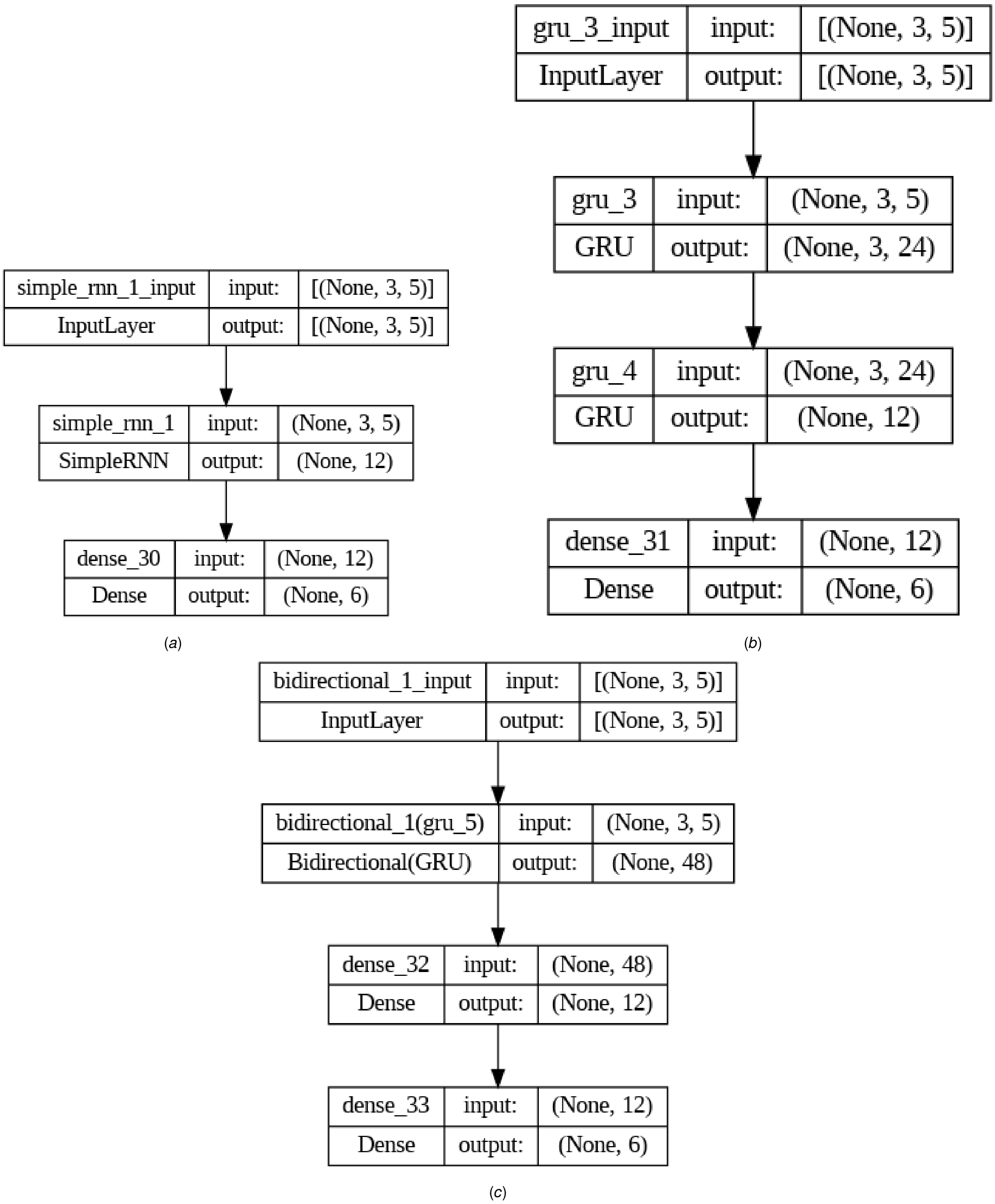


Fig. 14 (a) RNN architecture, (b) GRU Architecture, (c) Bidirectional GRU Architecture

List of Figures

List of Tables