



Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943



Efficient Human Activity Recognition using PCA Dimensionality Reduction and GWO-Enhanced LSTM

Israa Ramadhan Salman¹, Ali Abdulridha Rasheed², Saif Al-Deen H. Hassan^{3*}, Rasha Abed Hussein⁴, M.A. Abdelzاهر⁵

¹ College of Pharmacy, University of Misan, Maysan, Iraq

² Electronic Computing Center, University of Misan, Maysan, Iraq

³ Department Business Administrator, College of Administration and Economics, University of Misan, Maysan, Iraq

⁴ Al-Manara College for Medical Sciences, Maysan, Iraq

⁵ Environmental Science and Industrial Development Department, Faculty of Postgraduate Studies for Advanced Sciences, Beni-Suef University, Beni-Suef 62511, Egypt

ARTICLE INFO

Article history:

Received 23 June 2024

Received in revised form 25 November 2024

Accepted 30 November 2024

Available online 31 December 2024

Keywords:

Human activity recognition; principal component analysis; long short-term memory; Gray wolf optimization; dimensionality reduction; hyperparameter optimization; computational efficiency

ABSTRACT

Human activity recognition (HAR) is a critical task in various applications such as healthcare, smart homes, and security systems. The high dimensionality of the input data and the challenge of optimizing model parameters can hinder the development of accurate and efficient HAR systems. This research presents a novel method that combines Principal Component Analysis (PCA) for dimensionality reduction with Long Short-Term Memory (LSTM) networks. The LSTM network is optimized using the Grey Wolf Optimization (GWO) algorithm to enhance classification performance. Initially, PCA is employed to reduce the dimensionality of the input feature space, significantly reducing the number of features from 561 to 196 while preserving more than 99% of the original data variance. This step improves computational efficiency and reduces the risk of overfitting. The GWO algorithm is then used to fine-tune the LSTM network's hyperparameters, including the number of hidden units, learning rate, drop frequency, and batch size. This optimization ensures that the LSTM network effectively captures complex temporal dependencies in the activity data. The proposed method was rigorously tested and achieved a remarkable accuracy of 98.95%, demonstrating its robustness and efficacy in human activity recognition tasks. The integration of PCA and GWO improves the model's performance and enhances its generalization capability to new, unseen data. This approach offers a powerful and efficient solution for HAR, addressing fundamental challenges and paving the way for future advancements in the field.

1. Introduction

Human activity recognition (HAR) is a vital research area with applications in healthcare, smart environments, sports analytics and security systems. The ability to automatically identify and classify

* Corresponding author.

E-mail address: saif_aldeen@uomisan.edu.iq

<https://doi.org/10.37934/araset.54.2.317343>

human activities from sensor data enables a wide range of applications, including remote patient monitoring, behaviour analysis and activity-aware assistance. Despite significant advancements in sensor technology and machine learning algorithms, HAR remains a challenging task due to the complexity of human behaviour, variability in sensor data and the need for robust and accurate classification methods [1-3].

One of the primary challenges in HAR is the high dimensionality of the input data, often resulting from numerous sensors capturing multiple features. The significant feature space poses computational challenges and increases the risk of overfitting, leading the model to learn to memorize training data rather than generalize to new instances. Additionally, the variability and noise present in sensor data further complicate the recognition process, requiring sophisticated techniques to extract meaningful patterns and discriminate between different activities. These challenges underscore the need for advanced methods to handle high-dimensional data while maintaining high classification accuracy efficiently [4,5].

Furthermore, the optimization of model hyperparameters presents another significant challenge in HAR. Hyperparameters such as the number of hidden units in neural networks, learning rates and batch sizes play a crucial role in determining the performance and generalization capability of HAR models. However, finding the optimal combination of hyperparameters is a non-trivial task, often requiring extensive experimentation and computational resources. Inefficient hyperparameter tuning can lead to suboptimal model performance, slower convergence or even failure to learn meaningful patterns from the data. Addressing these challenges is essential for developing robust and accurate HAR systems that can effectively recognize human activities in real-world scenarios [6,7].

According to Wan *et al.*, [8], a smartphone inertial accelerometer-based architecture for Human Activity Recognition (HAR) is devised. The data undergo preprocessing steps including denoising, normalization and segmentation to extract valuable feature vectors. Finally, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), BLSTM, MLP and SVM models are applied to the UCI and Pamap2 datasets. A holistic deep learning-based architecture for activity recognition is proposed by Mutegeki *et al.*, [9]. The CNN-LSTM network used in this study is deep both spatially and temporally. Deep spatially refers to the convolutional layers in the CNN, which capture spatial features from the input data, such as sensor readings at different time intervals. Deep temporally refers to the LSTM layers, which are responsible for modelling the temporal dependencies and sequential nature of the data. This combination allows the network to learn both local and long-term patterns, which significantly contributes to the high performance of the model. As a result, a 99% accuracy is achieved on the iSPL dataset, an internal dataset and a 92% accuracy on the UCI HAR public dataset. According to Zhou *et al.*, [10], the focus is on improving HAR within the Internet of Health Things (IoHT) settings using deep learning methods. A deep Q-network (DQN) is introduced with a novel distance-based reward rule to boost learning efficiency in IoT environments. The DQN contributes by handling the challenge of weakly labelled sensor data through an intelligent auto-labelling scheme. This scheme utilizes the distance-based reward rule to improve the model's ability to label the data autonomously, thereby enhancing the training process. This improvement in labelling efficiency helps the system make better use of available data, leading to more accurate human activity recognition in IoHT settings. Furthermore, a classification approach utilizing LSTM is suggested to discern intricate patterns by extracting high-level features contextually from sequential motion data. The LSTM's role is crucial in capturing temporal dependencies in the sequential data, allowing the model to detect fine-grained activity patterns over time. This deep learning architecture, combining DQN and LSTM, ensures both efficient data labelling and precise activity recognition, significantly improving HAR performance in IoHT environments.

The transformer model originally designed for natural language processing and vision tasks, was modified by Dirgová *et al.*, [11] for analysing motion signals in time-series data. The self-attention mechanism within the transformer captures dependencies between signal values within a time series, allowing it to achieve performance comparable to state-of-the-art convolutional neural networks with long short-term memory. A hybrid feature selection process has been proposed by Ahmed *et al.*, [12], comprising both filter and wrapper methods. The process employs a sequential floating forward search (SFFS) to extract desired features, enhancing activity recognition. Subsequently, the features are utilized for training and testing a multiclass support vector machine (SVM) to construct nonlinear classifiers, employing the kernel trick. A HAR framework utilizing smartphone sensor data is presented by Mekruksavanich *et al.*, [13], primarily based on LSTM networks suitable for time-series analysis. A hybrid LSTM network, named 4-layer CNN-LSTM, is introduced to improve recognition accuracy. The 4-layer CNN-LSTM architecture has several advantages: the CNN layers perform direct mapping in the spatial representation of raw sensor data for effective feature extraction, while the LSTM layers leverage temporal dependencies, enhancing the extraction of features related to human activities. This combination allows the model to capture both spatial and temporal patterns more effectively than standard LSTM networks. The achieved accuracy on the UCI-HAR dataset with this architecture is 99.39%, which represents an improvement of up to 2.24% compared to prior state-of-the-art approaches. Bayesian optimization techniques are utilized for hyperparameter tuning of the LSTM networks. Evaluation of the HAR method is performed on the UCI-HAR dataset. Dahou *et al.*, [14] presents a novel HAR system that optimizes Convolutional Neural Network and Arithmetic Optimization Algorithm (AOA) to improve HAR performance while minimizing resource usage. The CNN extracts feature from input data and a modified version of AOA, Binary AOA (BAOA), selects optimal features. These features are then classified using a Support Vector Machine (SVM) for different activities. The proposed HAR model is evaluated on three public datasets: UCI-HAR, WISDM-HAR and KU-HAR datasets. According to Helmi *et al.*, [15], Deep Learning (DL) and Swarm Intelligence (SI) are combined to create a robust HAR system using wearable sensor data. The system employs a lightweight feature extraction approach, which integrates a residual convolutional network and a recurrent neural network (RCNN-BiGRU). The combination of DL and SI contributes to creating a robust HAR system by leveraging the strengths of both methodologies. DL methods, particularly through the use of residual convolutional networks and recurrent neural networks, excel at automatically extracting complex patterns and features from the raw sensor data. This reduces the need for manual feature engineering and enhances the model's ability to recognize nuanced human activities. Meanwhile, SI techniques, such as the marine predator algorithm (MPA), optimize the feature selection process by identifying the most relevant features, which can lead to improved model performance and reduced computational complexity. This synergy ensures that the HAR system not only effectively captures significant data patterns but also operates efficiently, making it suitable for real-time applications in various smart environments. Additionally, novel feature selection methods based on the marine predator algorithm (MPA) are developed to identify the optimal feature set. A federated learning system for Human Activity Recognition (HARFLS) is developed by Xiao *et al.*, [16]. A perceptive extraction network (PEN) is designed as the feature extractor for each user. The feature network, utilizing a convolutional block, is tasked with identifying local features within the HAR data, while the relation network, comprising a combination of LSTM and attention mechanism, focuses on uncovering global relationships inherent in the data. The performance of the system is evaluated using four widely used datasets: WISDM, UCI_HAR 2012, OPPORTUNITY and PAMAP2. According to Gil-Martín *et al.*, [17], wearable sensor data is used to analyse the primary motion characteristics of different types of movement. The study employs Convolutional and Recurrent Neural Networks for feature learning and classification. Evaluation is

conducted using recordings from the Pamap2 and Opportunity datasets with subject-wise cross-validation. A model named the Gated Recurrent Unit-Inception (GRU-INC) model is proposed by Mim *et al.*, [18], which utilizes an Inception-Attention-based approach incorporating the Gated Recurrent Unit (GRU). The temporal aspect of the model employs GRU along with Attention Mechanism (AM), while the spatial aspect utilizes the Inception module along with the Convolutional Block Attention Module (CBAM). According to Priyadarshini *et al.*, [19], various machine learning and deep learning algorithms are explored for HAR, including Random Forest (RF), Decision Trees (DT), K-Nearest Neighbors (k-NN), CNN, LSTM and Gated Recurrent Units (GRU). Optimization techniques are introduced to enhance the performance of CNN, LSTM and GRU models, utilizing Stochastic Gradient Descent (SGD) along with optimizers such as Adam and RMSProp. Evaluation metrics include Accuracy and F-1 score. According to Abdel-Basset *et al.*, [20], a novel framework for human activity recognition is introduced, leveraging raw readings from a blend of smartphone sensors, including accelerometer, gyroscope, magnetometer and Google Fit activity tracking module. The framework employs a deep recurrent neural network (DRNN) trained on an extensive dataset containing five activity classes from 12 individuals. The study by Iloga *et al.*, [21] presents a Hidden Markov Model (HMM)-based approach for Human Activity Recognition. Meta-data extracted from these models are incorporated as components of the feature vectors. Classification experiments 2 living and 8 types of falls. According to Bijalwan *et al.*, [22], data collection involves various activities utilizing an IMU sensor equipped with a three-axis accelerometer, three-axis gyroscopes and a 3° magnetometer. The gathered data undergo preprocessing using various filters and cubic spline techniques. Following preprocessing, the data is categorized into seven activities. Subsequently, the data is classified using four deep learning models: deep neural network, bidirectional-long short-term memory (BLSTM), CNN and CNN-LSTM.

The issue of classifying motion signals produced by multiple wearable sensors for human activity recognition and sensor localization is tackled by Lawal *et al.*, [23]. Frequency images are utilized as the input for the proposed two-stream CNN to predict both the human activity and the location of the sensor generating the activity signal. A study by Chen *et al.*, [24] introduces a two-stage genetic algorithm-based feature selection algorithm known as GFSFAN. The study evaluates the performance of Human Activity Recognition using six classifiers to assess the impact of selected feature subsets from different feature selection algorithms. The Contrastive Predictive Coding (CPC) framework is introduced by Haresamudram *et al.*, [25] to human activity recognition, capturing the temporal structure of sensor data streams. Its effectiveness for enhanced HAR is demonstrated through various experimental evaluations on real-life recognition tasks. The pre-training based on CPC is self-supervised and the learned representations derived from it can be incorporated into conventional activity chains. In a study by Zhu *et al.*, [26], a hybrid classifier is introduced, that integrates CNNs and recurrent neural networks (RNNs) to extract spatial-temporal patterns. The performance of the classifier is assessed through validation using K-fold cross-validation (CV) and leave-one-person-out (L1PO) methods. According to Tong *et al.*, [27], a novel deep learning model for HAR utilizing inertial sensors is proposed. A deep learning model called Bidirectional-Gated Recurrent Unit-Inception (Bi-GRU-I) is devised to enhance accuracy and reduce parameter count. Finally, comparative experiments are conducted with other methods on three datasets: the self-collected CATP dataset, the widely used Wireless Sensor Data Mining (WISDM) dataset and the University of California, Irvine (UCI-HAR) dataset. In a study by Shavit *et al.*, [28], an activity recognition model based on Transformers is introduced, providing an enhanced and comprehensive framework for learning activity recognition tasks. For evaluation, multiple datasets comprising over 27 hours of inertial data recordings from 91 users are utilized. While Li *et al.*, [29] introduces Meta-HAR, a federated representation learning framework. Within this framework, a signal embedding network is meta-

learned in a federated manner and the resulting signal representations are subsequently utilized by personalized classification networks at individual users for activity prediction. The shared embedding network undergoes training using a Model-Agnostic Meta-learning framework. The objective of Ferrari *et al.*, [30] is to assess whether incorporating personalization into both traditional and deep learning methods yields superior performance compared to classical approaches lacking personalization. AdaBoost was selected as the traditional machine learning technique, while CNN was chosen for deep learning, given their demonstrated effectiveness. This implies that traditional deep learning may prove more effective when working with a sizable and diverse dataset. In a study by Ihianle *et al.*, [31], a deep learning multi-channel architecture is introduced, comprising a combination of CNN and Bidirectional long short-term memory (BLSTM). The model benefits from the direct mapping and abstract representation of raw sensor inputs by the CNN layers, facilitating feature extraction at various resolutions. According to Park *et al.*, [32], a deep learning-based HAR model named MultiCNN-FilterLSTM is introduced to support resource-efficient IoT systems. This model combines a multihead CNN with a LSTM through a residual connection, enabling efficient processing of feature vectors in hierarchical order.

The current research landscape in human activity recognition has seen significant advancements, with numerous studies focusing on developing novel algorithms and techniques to improve classification accuracy and robustness. However, a notable gap exists in the literature regarding the integration of dimensionality reduction and hyperparameter optimization techniques within a unified framework for HAR. While individual methods of dimensionality reduction and optimization algorithms for tuning hyperparameters have been explored independently, there is limited research on their combined application in the context of HAR. This study seeks to bridge this gap by proposing a novel method that integrates PCA for dimensionality reduction with LSTM networks optimized using GWO, aiming to enhance classification performance while addressing the challenges of high dimensionality and hyperparameter tuning.

The proposed methodology builds upon powerful algorithms in dimensionality reduction and hyperparameter optimization, leveraging the strengths of PCA and GWO to address key challenges in HAR. PCA serves as an effective technique for reducing the dimensionality of the input feature space, allowing for more efficient processing and improved generalization capability of the model. By retaining the most informative features while discarding redundant ones, PCA facilitates better model interpretation and performance. Meanwhile, GWO offers a robust optimization framework for fine-tuning the hyperparameters of LSTM networks, ensuring that the model is effectively trained to capture complex temporal dependencies in the activity data. Through the integration of these techniques, the proposed method aims to provide a comprehensive solution for enhancing the accuracy and efficiency of HAR systems.

In the subsequent sections of this research, we will present a detailed description of the proposed methodology, including the basic concepts and implementation steps. We will first introduce the PCA-based dimensionality reduction technique and its application to preprocess the input data. Next, we will discuss the LSTM network, GWO algorithm and its utilization for optimizing the hyperparameters of LSTM networks. After that, the dataset used for simulation and the evaluation metrics employed for assessment are explained. Following that, we will provide experimental results demonstrating the effectiveness of the proposed method in improving classification performance on benchmark UCI HAR datasets. Next, the comparison of the proposed method with some other methods of the literature is presented. Finally, we will conclude with the implications of our findings of our proposed method.

2. Methodology

2.1 Principal Component Analysis

PCA is a flexible statistical technique used to distil a data table, consisting of cases and variables, down to its fundamental elements, known as principal components. Several multivariate methods are simple variations of PCA. One way to modify PCA is by changing the distance function, which alters the measure of total variance. Another variation involves assigning different weights to the cases so that some cases have more influence on the PCA solution than others. In PCA, the distances between the projected points approximate the Euclidean distances between the points in the original space. The Euclidean distance between points i and i' is defined as:

$$d(i, i') = \sqrt{\sum_j (y_{ij} - y_{i'j})^2} \quad (1)$$

where y_{ij} and $y_{i'j}$ refer to the standardized data. If the original data are denoted by x_{ij} and standardization is performed by subtracting the mean \bar{x}_j and dividing by the standard deviation s_j , then:

$$y_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \quad (2)$$

and the distance formula becomes:

$$d(i, i') = \sqrt{\sum_j \left(\frac{x_{ij} - x_{i'j}}{s_j}\right)^2} \quad (3)$$

This is called the standardized Euclidean distance, in this distance the inverses of the variances $w_j = \frac{1}{s_j^2}$ can be considered as weights on the variables.

A variant of PCA is correspondence analysis, which is generally applied to two-way cross-tabulations, general frequency data or data in the form of percentages. In correspondence analysis, the relative values of the data are of interest, such as rows divided by their row totals, called profiles. The distances between profiles, the chi-square distances, have a form similar to the standardized Euclidean distance. Specifically, both distance measures quantify the difference between two points in a multi-dimensional space. The standardized Euclidean distance adjusts the raw differences by the standard deviation of each dimension, effectively normalizing the scale of the data. In contrast, the chi-square distance measures the divergence between two probability distributions, focusing on the ratio of the differences relative to the total counts. Despite their different applications, both distances aim to assess similarity or dissimilarity between datasets and share a foundational mathematical structure that enables the comparison of multivariate data. This distance enoting the (row) profiles by r_{ij} :

$$d(i, i') = \sqrt{\sum_j \left(\frac{r_{ij} - r_{i'j}}{c_j}\right)^2} \quad (4)$$

where c_j is the j th element of the average profile. Therefore, for relative frequency data, the mean profile element c_j replaces the variance s_j^2 in the equation. This implies that the weights for the variables are the inverses $1/c_j$. The reason the mean profile element replaces the variance in this case

is that, in correspondence analysis, we are primarily interested in relative values (i.e., proportions) rather than absolute deviations. The mean profile element represents the average distribution of each variable, so replacing the variance with c_j emphasizes the relative importance of deviations from this average. This affects the interpretation of the weights by shifting the focus from variability within each variable (as measured by the variance) to the contribution of each variable relative to the overall data structure. Consequently, variables with larger mean profile elements contribute less to the distance calculation, since their relative deviations are smaller, aligning with the principle of correspondence analysis where we are more concerned with proportional differences. Correspondence analysis focuses on the relative values of the data, such as row profiles, which are calculated by dividing rows by their row totals.

Eigenvalues and eigenvectors play a crucial role in PCA. Eigenvalues represent the amount of variance captured by each principal component and eigenvectors indicate the direction of these components. When the covariance matrix of the standardized data is computed, the eigenvalues and corresponding eigenvectors are derived from this matrix. The principal components are then identified as the eigenvectors with the largest eigenvalues, capturing the most significant patterns in the data illustrated in Figure 1:

- i. The rows of data, which can be optionally standardized, along with their mean or centroid (C), define points in a multidimensional space. Standardization is optional because it depends on whether the scales of the variables are comparable. If the variables are measured on different scales, standardization is necessary to ensure that no single variable dominates the analysis due to its scale. In contrast, if all variables are already on the same scale, standardization may not be required. The implication of not standardizing the data when needed is that variables with larger magnitudes could disproportionately influence the principal components, leading to skewed results. Conversely, unnecessary standardization may dilute meaningful variability between variables that are naturally measured on the same scale.
- ii. The first two dimensions from the singular value decomposition identify the optimal two-dimensional plane that fits the data points in terms of least-squared distances between the plane and the points. This plane includes the centroid (C), which becomes the origin of the PCA display and represents the averages of the variables.
- iii. Each multidimensional data point forms a right-angled triangle with its projection onto the plane and the centroid. The total variance is fixed and equals the average sum of squared distances of the points to the centroid. Maximizing the average squared distances in the plane (maximizing variance) is equivalent to minimizing the average squared distances from the points to the plane (minimizing fit) [33]. In the context of PCA, maximizing the variance ensures that the principal components capture the most significant patterns or directions of data spread. The goal of PCA is to project the data into a lower-dimensional space while retaining as much variability (or information) as possible. When the variance is maximized, the points are spread out as much as possible along the principal components, meaning the most important features of the data are preserved. This is equivalent to minimizing the fit because, in PCA, "fit" refers to how closely the points adhere to their projections onto the lower-dimensional plane. By minimizing the average squared distances from the points to the plane, we ensure that the plane fits the data well and that the components provide the best possible representation of the data's structure. This dual objective aligns with PCA's goal of finding the projection that best summarizes the data with minimal loss of information.

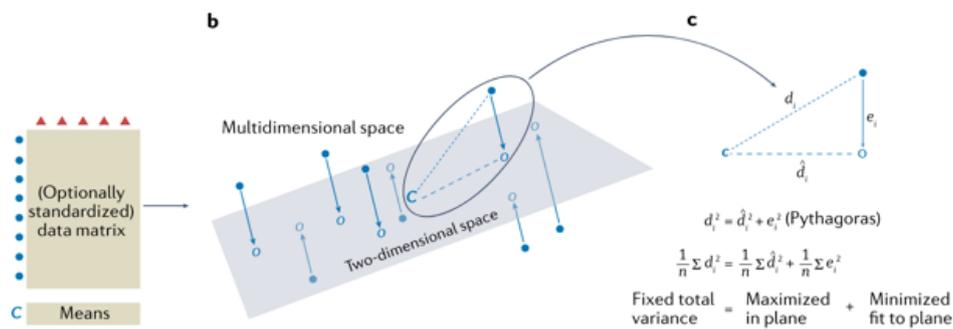


Fig. 1. Diagrammatic representation of dimensionality reduction in PCA [33]

2.2 Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) frequently have trouble retaining data over extended intervals of time. This problem is solved in LSTM models by including memory cells with gated mechanisms. With the help of these gates, the model can decide which information is relevant to keep or discard.

- i. **Forget Gate:** This gate decides which information to discard from the LSTM memory. It uses a sigmoid function to compute a value, f_t , ranging between 0 and 1. This value determines how much of the previous information h_{t-1} and the current input x_t should be kept or discarded. Mathematically, this is represented as Eq. (5):

$$f_t = \sigma(W_{fh}[h_{t-1}] + W_{fx}[x_t], b_f) \quad (5)$$

- ii. **Input Gate:** The input gate decides whether new information should be added to the LSTM memory. It comprises two layers: a sigmoid layer and a hyperbolic tangent (tanh) layer. The sigmoid layer determines the update signal i_t , indicating which parts of the memory to update. The tanh layer generates a vector \tilde{c}_t of candidate values to be added to the memory. The combination of these layers determines the memory update, computed as:

$$i_t = \sigma(W_{ih}[h_{t-1}] + W_{ix}[x_t], b_i) \quad (6)$$

$$\tilde{c}_t = \tanh(W_{ch}[h_{t-1}] + W_{cx}[x_t], b_c) \quad (7)$$

The updated memory c_t (Eq. (8)) is calculated by combining the old memory c_{t-1} with the new candidate value $i_t \tilde{c}_t$:

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (8)$$

- iii. **Output Gate:** The output gate determines which part of the LSTM memory contributes to the output. It starts with a sigmoid layer to compute the output gate signal o_t indicating the relevance of the memory. The sigmoid function produces a value between 0 and 1, which determines the proportion of the current cell state that should influence the final output. The tanh function then maps the memory cell value (c_t) to a range between -1 and 1. The output gate signal o_t (from the sigmoid layer) is multiplied with the tanh-scaled cell state. This combination ensures that only the relevant portion of the current memory cell, as determined by the output gate, is passed to the final output h_t . This is followed by

a tanh function that maps the values between -1 and 1 and then multiplies with the output of another sigmoid layer to produce the final output h_t :

$$o_t = \sigma (W_{oh}[h_{t-1}] + W_{ox}[x_t], b_o) \quad (9)$$

$$h_t = o_t \tanh (c_t) \quad (10)$$

The forget gate, input gate and output gate are shown from left to right in Figure 2, which depicts the internal architecture of an LSTM unit [34].

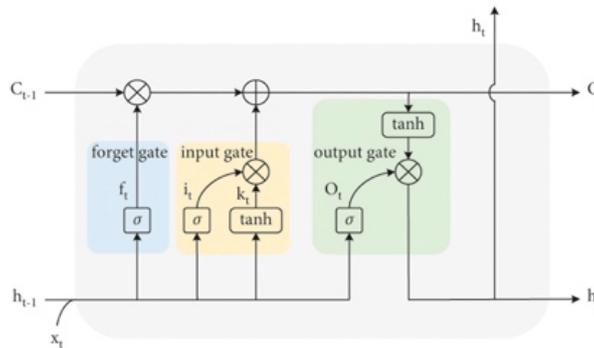


Fig. 2. The internal architecture of the basic LSTM network unit [35]

2.3 Gray Wolf Optimization Algorithm

Gray Wolf Optimization, which is inspired by the social structure and hunting behaviour of gray wolves in nature. The algorithm is based on the social hierarchy of gray wolves, with four types: alpha, beta, delta and omega, each representing a different leadership role within the pack. The GWO algorithm comprises three main steps: encircling, searching and attacking prey.

At the top of the hierarchy is the alpha wolf, who leads and coordinates the pack's hunting efforts. Beta wolves, in the second tier, relay feedback and information to the alpha from other wolves. Delta wolves, in the next level, control the omega wolves, which are at the lowest rank.

Eq. (11) calculates the distance of each wolf from the alpha, beta and delta wolves based on their positions (denoted as X) in the search space:

$$\begin{aligned} D_{alpha} &= |C_1 \cdot X_{alpha} - X|, \\ D_{beta} &= |C_2 \cdot X_{beta} - X|, \\ D_{delta} &= |C_3 \cdot X_{delta} - X| \end{aligned} \quad (11)$$

To determine the next position of the current wolf, the values of X_1 , X_2 and X_3 are computed using Eq. (12):

$$\begin{aligned} X_1 &= X_{alpha} - A_1 \cdot D_{alpha}, \\ X_2 &= X_{beta} - A_2 \cdot D_{beta}, \\ X_3 &= X_{delta} - A_3 \cdot D_{delta} \end{aligned} \quad (12)$$

The parameters A , a and C , which govern the algorithm's behaviour, are determined using Eq. (13). The variables r_1 and r_2 are randomly generated values within the range of 0 to 1. C is a parameter that enhances the GWO algorithm's exploration capability:

$$\begin{aligned} A &= 2a \cdot r_1 - a, \\ C &= 2 \cdot r_2 \end{aligned} \tag{13}$$

The next position of every wolf in the population is adjusted according to Eq. (14):

$$X(t + 1) = (X_1 + X_2 + X_3)/3 \tag{14}$$

When $|A|$ is less than 1, it indicates that the wolf is attached to the prey. Conversely, if $|A|$ exceeds 1, the wolf distances itself from the prey and seeks out more suitable targets. The search behaviour of gray wolves is influenced by the positions of the alpha (α), beta (β) and delta (δ) wolves. The A and C vectors play a crucial role in balancing exploitation and exploration. A is a control parameter that modulates the degree of exploration and exploitation in the search process. The magnitude of A indicates the strength of attraction or repulsion from the prey: a value less than 1 suggests the wolf is following the prey closely, while a value greater than 1 implies the wolf is moving away from it to explore other areas. The purpose of the random values of A is to introduce variability in the wolves' movements, helping to distinguish between their roles as followers (staying close to the prey) and explorers (searching for new targets). This randomness allows the algorithm to avoid premature convergence on local optima and encourages a more thorough search of the solution space. Random values of C , ranging from 0 to 2, prevent the algorithm from getting stuck in local optima. A higher value of C increases the search process's randomness, making it harder for wolves to stay close to their prey [36]. Figure 3 illustrates the algorithm's behaviour.

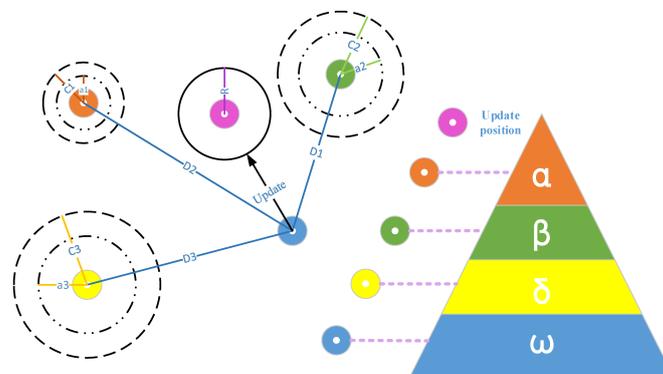


Fig. 3. The diagram of the GWO algorithm [37]

The main focus throughout the process is on adjusting the A and C vectors, with most parameters emphasizing either exploitation or exploration. The GWO algorithm concludes once the specified benchmark is achieved, yielding the best location for the alpha wolf.

Figure 4 illustrates the flowchart of the Gray Wolf Optimization algorithm.

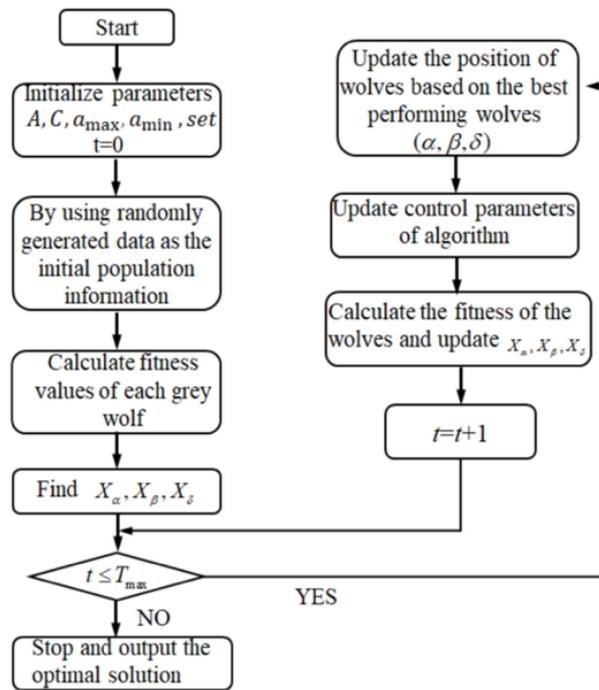


Fig. 4. The flowchart of the GWO algorithm [38,39]

2.4 Stages of the Proposed Method

In this section, we detail the methodology employed to develop a robust human activity recognition system. The proposed method integrates Principal Component Analysis (PCA) for dimensionality reduction and Long Short-Term Memory (LSTM) networks optimized using the Grey Wolf Optimization (GWO) algorithm. This hybrid approach is designed to enhance classification performance by focusing on the most informative features and fine-tuning the model's hyperparameters.

The initial phase of our method involves reducing the dimensionality of the input feature space using PCA. The original dataset has a large number of features, which can lead to overfitting and increased computational complexity. PCA is utilized to transform the feature space, concentrating the data's variance in the first few principal components. This transformation effectively reduces the dimensionality while preserving the most informative original content.

To achieve this, we calculate the eigenvalues of the covariance matrix of the transformed features and normalize them to form a vector indicating each feature's contribution to the data variance. The covariance matrix is a square matrix that describes the variance and covariance between different features in the dataset. Each element of the matrix represents the covariance between a pair of features, indicating how much two variables change together. The eigenvalues of the covariance matrix are crucial for PCA because they represent the amount of variance captured by each principal component. A higher eigenvalue indicates that the corresponding principal component accounts for a larger portion of the variance in the data, allowing us to identify the most significant features for dimensionality reduction. This is essential in PCA, as it helps in selecting the components that contribute most to the variability in the dataset. By selecting the top principal components that account for more than 99% of the total variance, these findings reduce the feature set from 561 to 196 features. This reduction not only streamlines the dataset but also ensures that the most significant features are retained for the subsequent modelling phase.

The application of PCA offers several advantages, such as reduced overfitting and enhanced accuracy and generalization. Moreover, it enhances computational efficiency by reducing the number of features the model needs to process, which accelerates the training phase and reduces the risk of overfitting. Additionally, by focusing on the most informative features, PCA helps improve the model's accuracy and generalization capability. The transformed features form a new, more compact representation of the data, which is then used as input for the LSTM network. This step is crucial in addressing the high dimensionality of the data, a common challenge in human activity recognition, by ensuring that the model focuses on the most relevant features while discarding redundant or irrelevant ones.

Following dimensionality reduction, we focus on optimizing the LSTM network's hyperparameters to improve its performance. The hyperparameters tuned include the number of hidden units, learning rate, learning rate drop frequency and batch size. Properly setting these parameters is crucial as they significantly influence the network's ability to learn and generalize from the data.

We employ the Grey Wolf Optimization algorithm to determine the optimal hyperparameter settings. GWO mimics the leadership hierarchy and hunting mechanism of grey wolves in nature, making it an effective approach for solving complex optimization problems. The optimization process involves setting appropriate ranges for each hyperparameter to ensure a thorough and efficient search.

The GWO algorithm iteratively searches within these ranges to identify the combination that minimizes the error on the validation set. The process balances the thoroughness of the search with computational efficiency. During the optimization phase, the LSTM network is trained for a limited number of epochs per iteration to expedite the process. Once the optimal hyperparameters are identified, the final model is trained using these settings over an extended period to ensure thorough learning and convergence.

Optimizing the LSTM network's hyperparameters addresses several critical challenges in human activity recognition, including mitigating overfitting, ensuring proper model convergence, enhancing the model's ability to generalize to unseen data and efficiently learning from temporal patterns without being hindered by noisy or redundant information. Poorly chosen hyperparameters can lead to suboptimal performance, slow convergence or even failure to learn meaningful patterns. The use of GWO for hyperparameter optimization ensures that the LSTM network is finely tuned to capture the complex temporal dependencies in the data, enhancing its predictive accuracy and robustness.

Furthermore, human activity recognition systems must generalize well to new, unseen data to be practically useful. The combination of PCA and GWO not only improves the model's training performance but also enhances its generalization capability. This is critical for real-world applications where the model will encounter a wide variety of activities and conditions. The robustness of the proposed method is also reflected in its ability to handle noisy and incomplete data. The PCA step helps in filtering out noise by focusing on the principal components, while the optimized LSTM architecture, determined by GWO, is adept at learning from the data despite imperfections. Figure 5 illustrates the flowchart of the proposed method based on the mentioned stages.

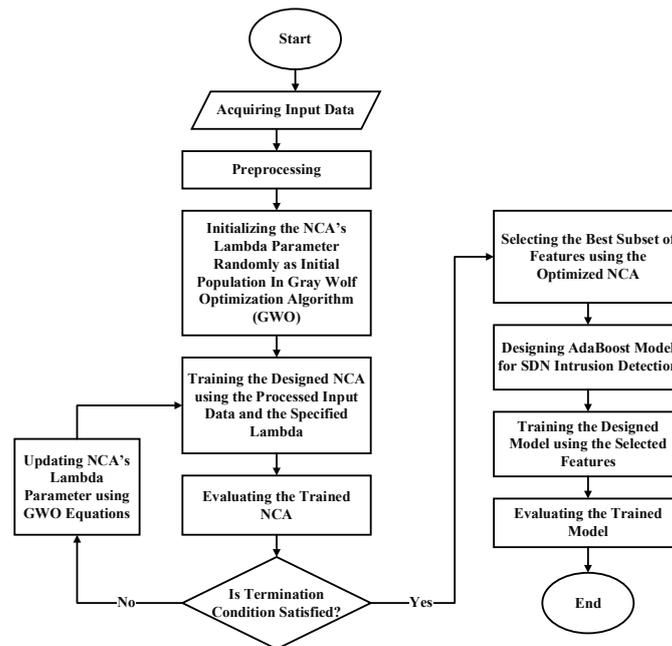


Fig. 5. Flowchart of the proposed method

3. Dataset

In this study, we utilize the UCI HAR dataset for the evaluation of the proposed method. The Human Activity Recognition dataset (UCI HAR) was constructed from recordings of 30 individuals performing daily activities while carrying a smartphone equipped with inertial sensors on their waist. The participants, aged between 19 and 48 years, engaged in six activities: walking, walking upstairs, walking downstairs, sitting, standing and lying. A Samsung Galaxy S II smartphone, worn at the waist, recorded 3-axis linear acceleration and 3-axis angular velocity at a consistent rate of 50Hz. The data collection was video-recorded to facilitate manual labelling.

The dataset was divided into training and test sets, with 7352 subjects' data allocated for training and 2947 samples for testing. Sensor data from the accelerometer and gyroscope underwent noise filtering and were sampled using fixed-width sliding windows of 2.56 seconds with a 50% overlap, resulting in 128 readings per window. This overlap allows for a more granular representation of the data, ensuring that critical transitional states between activities are not lost. By using overlapping windows, we can maintain continuity in the data, providing the model with more samples that contain variations and nuances inherent in the activities being performed. Additionally, the overlap enhances the ability of the model to learn and generalize by exposing it to multiple views of the same activity, thus improving the robustness of the feature extraction process and leading to better performance in recognizing distinct activities. The accelerometer's signals, which include both gravitational and body motion components, were decomposed using a Butterworth low-pass filter into body acceleration and gravity components. A cutoff frequency of 0.3 Hz was applied to isolate the gravitational force, presumed to consist solely of low-frequency components.

The features selected for this dataset come from the accelerometer and gyroscope 3-axial raw signals (time-domain signals) tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. They were then filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Specifically, "tAcc-XYZ" refers to the time-domain accelerometer data for the three spatial axes (X, Y and Z), while "tGyro-XYZ" represents the time-domain gyroscope data for the same axes. Similarly, the

acceleration signal was separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). The magnitude of these three-dimensional signals was calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

A Fast Fourier Transform (FFT) was then applied to some of these signals, producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag and fBodyGyroJerkMag ('f' indicates frequency domain signals).

From each window, features were extracted by computing various time and frequency domain variables, producing a 561-feature vector for each record. The dataset includes the triaxial acceleration (total and body), triaxial angular velocity, activity labels and an identifier for the subject. Additionally, it provides labels for postural transitions and the complete raw inertial signals prior to preprocessing.

The signals used to estimate variables of the feature vector include tBodyAcc-XYZ, tGravityAcc-XYZ, tBodyAccJerk-XYZ, tBodyGyro-XYZ, tBodyGyroJerk-XYZ, tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag, fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccMag, fBodyAccJerkMag, fBodyGyroMag and fBodyGyroJerkMag.

The set of variables estimated from these signals includes several statistical measures. First, mean() calculates the mean value of the signal. The std() function computes the standard deviation, while mad() assesses the median absolute deviation. Additionally, max() identifies the largest value in the array and min() finds the smallest value. The sma() function measures the signal magnitude area.

Furthermore, energy() calculates the energy measure, defined as the sum of the squares divided by the number of values. The iqr() function provides the interquartile range and entropy() measures signal entropy. For autoregressive modeling, arCoeff() computes autoregression coefficients with a Burg order equal to 4.

The correlation() function finds the correlation coefficient between two signals. In the frequency domain, maxInds() identifies the index of the frequency component with the largest magnitude. The meanFreq() function determines the weighted average of the frequency components, yielding a mean frequency. Finally, skewness() and kurtosis() calculate the skewness and kurtosis of the frequency domain signal, respectively.

Additional functions include bandsEnergy(), which measures the energy of a frequency interval within the 64 bins of the FFT of each window and angle(), which calculates the angle between two vectors.

Additional vectors obtained by averaging the signals in a signal window sample are used in the angle() variable, including gravityMean, tBodyAccMean, tBodyAccJerkMean, tBodyGyroMean and tBodyGyroJerkMean [40].

3.1 Evaluation Metrics

In the context of Human Activity Recognition (HAR), evaluating the performance of classification models is crucial to understanding how well the model can predict different activities. Common evaluation metrics include accuracy, precision, recall and the F1 score. Here's how these metrics apply to HAR:

- i. Accuracy: Accuracy is the ratio of correctly predicted activities to the total number of predictions. It gives a general idea of the overall performance of the model. However, accuracy can be misleading in cases of imbalanced datasets where some activities are more frequent than others.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

- ii. Precision: Precision measures the proportion of true positive predictions among all positive predictions (true positives and false positives). In HAR, precision for an activity (e.g., walking) indicates how many of the predicted instances of walking were actually walking.

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

- iii. Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of true positives among all actual positives (true positives and false negatives). In HAR, recall for an activity (e.g., walking) indicates how many of the actual instances of walking were correctly identified by the model. For instance, if there are 100 actual instances of walking (true positives + false negatives) and the model correctly identifies 80 of them as walking (true positives), while it fails to identify 20 (false negatives), the recall would be calculated as follows:
Recall = True Positives / (True Positives + False Negatives) = 80 / (80 + 20) = 80 / 100 = 0.8 or 80%. This means that the model correctly identifies 80% of the actual walking instances.

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

- iv. F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both concerns, especially useful when the dataset has imbalanced classes. In HAR, the F1 score for an activity (e.g., walking) provides a balance between the precision and recall of predicting that activity. The F1 score is particularly useful in cases of imbalanced datasets because traditional accuracy metrics can be misleading. For example, if a model predicts the majority class correctly most of the time, it can achieve high accuracy without effectively recognizing the minority class. The F1 score, by considering both precision and recall, ensures that the model is not only identifying the majority class but also performing well on the minority class, thereby providing a more comprehensive evaluation of its performance.

$$F_1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (18)$$

In a HAR system trained to recognize walking, walking upstairs and walking downstairs, a high F1 score for 'walking upstairs' would indicate that the model is reliably able to distinguish between walking upstairs and other activities. A high accuracy suggests overall good performance, while high precision and recall for individual activities indicate the model's reliability in predicting those specific activities. The F1 score is particularly useful when the goal is to find a balance between precision and recall, ensuring that the model performs well in identifying true activities without being misled by false positives or false negatives.

4. Results

4.1 Simulation Results

This section provides in-depth findings derived from evaluating the proposed approach for human activity recognition. The proposed methodology was simulated using MATLAB 2023b on a system featuring the following specifications: an Intel Core i7 13650HX CPU running at 2.6 GHz with 24M Cache, 16 GB of RAM and graphical memory provided by an NVIDIA RTX 4050 with 6GB capacity.

4.1.1 Dimensionality reduction results

The initial step in implementing the proposed method, following pre-processing, involves reducing the dimensionality of the input data using Principal Component Analysis (PCA). The input dataset originally comprises 561 features—a substantial number that can lead to overfitting and complicate the training process. Therefore, dimensionality reduction is essential to enhance computational efficiency and model performance. PCA transforms the coordinate axes of the input space, mapping the data to a new space where most of the initial information is concentrated in the first few features.

By applying PCA, the variance in the data is reallocated such that the first principal components capture the most significant aspects of the data variability. This transformation not only reduces the complexity but also mitigates the risk of overfitting by focusing on the most informative features. Figure 6 demonstrates the class discrimination using the first three features of the data, both before and after applying PCA. As illustrated, the data's discrimination improves significantly post-PCA application, indicating a more distinct separation among different classes.

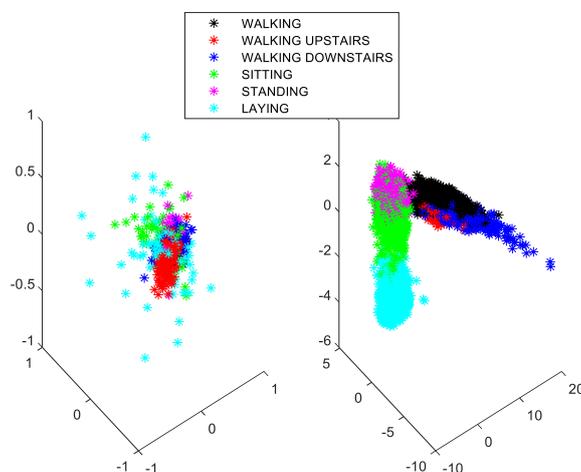


Fig. 6. The discrimination of different classes, before and after applying the PCA

The proposed method, however, is not limited to these three features. To determine the specific number of features to retain from the transformed dataset, we closely examine the amount of information captured in the new space. Our objective is to retain more than 99% of the original information to ensure that the most critical aspects of the data are preserved. We achieve this by calculating the eigenvalues for each new feature and normalizing them by dividing by the total sum of eigenvalues. This normalization process results in a vector where the sum of values equals 1, indicating the proportion of information each feature contributes to the total data variance.

Next, we sort this vector in descending order and compute its cumulative sum. The feature corresponding to the first element of this sorted vector has the highest eigenvalue, contributing the most to the data's information content. The cumulative sum of these values allows us to identify the point at which we capture the desired amount of information. To select an appropriate number of features, we retain features up to the point where the cumulative sum reaches 0.995, representing 99.5% of the original information. This systematic approach ensures which the current study is included the most relevant features while excluding less informative ones.

By selecting these features and removing the others, we effectively reduced the dataset to 196 features out of the original 561. This reduced feature set is then used to train the LSTM model, ensuring that the model is trained on the most informative and significant aspects of the data. This reduction not only speeds up the training process but also enhances the model's ability to generalize to new data by focusing on the most critical features. Figure 7 illustrates this feature selection process using the sorted cumulative eigenvalues.

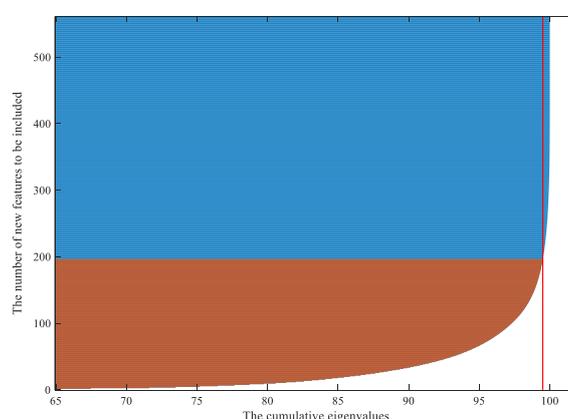


Fig. 7. The feature selection procedure using the sorted cumulative eigenvalues (the red line is the considered threshold)

4.1.2 LSTM optimization results

Following the reducing dimensionality of the initial dataset, we proceed to optimize the hyperparameters of the Long Short-Term Memory (LSTM) network, specifically the number of hidden units, learning rate, learning rate drop frequency and batch size. This optimization is carried out using the Grey Wolf Optimization (GWO) algorithm. Proper tuning of these parameters is crucial as they significantly impact the performance of the network.

The first step involves configuring the settings for the GWO algorithm, including the upper and lower limits for the decision variables, the number of population members and the maximum number of iterations. Given the specific ranges for each decision variable (LSTM hyperparameters), these values are selected to ensure a comprehensive search space. The range for the number of hidden units is set between (2, 150), the learning rate between (0.0001, 0.01), the learning rate drop frequency between (2, 80) and the batch size between (2, 120).

Adjustable parameters such as the number of population members and the maximum number of iterations play a pivotal role in the optimization process. Larger values allow for a more thorough exploration of the search space, whereas smaller values expedite the process. Considering that each iteration involves training the LSTM network and evaluating its error, a balance is needed to prevent excessively long optimization times. Given the relatively small search space defined by our parameter ranges, we set the maximum number of iterations to 9 and the number of population members to 3.

To optimize the speed of this process without sacrificing accuracy, we limit the number of training epochs for the LSTM network to 5 during the hyperparameter optimization phase. Although fewer epochs accelerate the process, this may risk the network not converging to the global optimum, potentially impacting accuracy. It is important to note that this phase focuses solely on comparing different hyperparameter settings and final training is conducted with the optimal hyperparameters identified.

An additional parameter to facilitate fast network convergence is the choice of the optimization algorithm. For this study, we utilize the well-regarded RMSPROP algorithm, known for its rapid convergence. Figure 8 illustrates the convergence curve of the LSTM hyperparameter optimization process using the GWO algorithm.

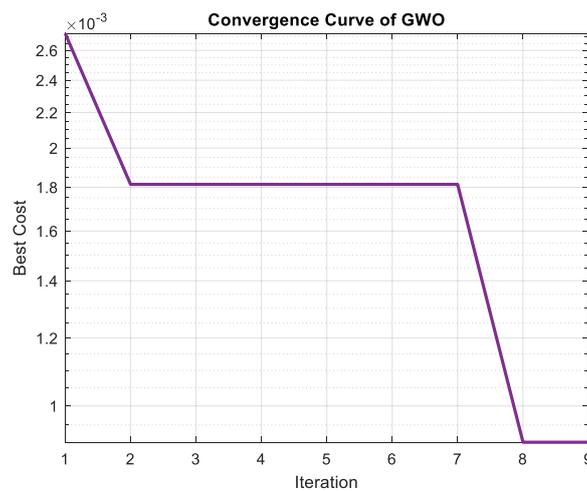


Fig. 8. The convergence curve of the GWO algorithm in optimizing LSTM hyperparameters

Table 1 presents the optimal values obtained through this process.

Table 1
 The optimized hyperparameters of LSTM

Parameter	Value
The number of hidden units	135
Learning rate	0.0035
Learning rate drop frequency	40
Batch size	100

Finally, the network is trained using the identified optimal parameters over 80 epochs. Figure 9 depicts the convergence curve of the final training session. The subsequent section evaluates the performance of the trained network using the proposed approach.

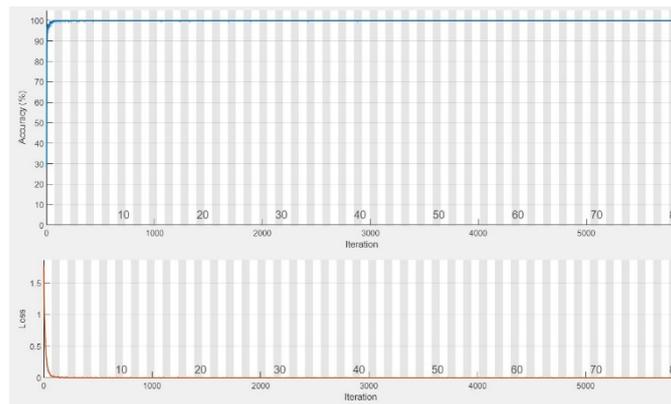


Fig. 9. The convergence curve of training LSTM using RMSPROP algorithms

4.1.3 Classification performance

In this section, we present and analyse the performance of the proposed method using various evaluation metrics. The classification performance is illustrated through confusion matrices, ROC curves and bar charts of evaluation metrics for both training and testing datasets.

A confusion matrix provides a comprehensive way to visualize the performance of a classification model. It displays the true positive, false positive, true negative and false negative counts, allowing for a detailed assessment of the model's accuracy. The matrix is particularly useful for understanding the types of errors the model is making.

Figure 10 and Figure 11 display the confusion matrices for the training and testing datasets. The rows represent the actual classes, while the columns represent the predicted classes. High values along the diagonal indicate correct predictions, whereas off-diagonal values represent misclassifications.

The confusion matrix for the training dataset (Figure 10) shows that all of the instances are correctly classified. This suggests that the model has effectively learned the patterns in the training data.

	1	2	3	4	5	6	
1	1226 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	1073 14.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	986 13.4%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	1286 17.5%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1374 18.7%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1407 19.1%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	6	

Fig. 10. Evaluating the proposed method using the confusion matrix for the training dataset

The confusion matrix for the testing dataset (Figure 11) indicates the model's performance on unseen data, with only 31 misclassified samples. This is expected due to the model's exposure to new data during testing, yet the overall high values along the diagonal indicate good generalization capability.

Output Class	1	2	3	4	5	6	
1	495 16.8%	18 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.5% 3.5%
2	1 0.0%	452 15.3%	1 0.0%	3 0.1%	0 0.0%	0 0.0%	98.9% 1.1%
3	0 0.0%	0 0.0%	419 14.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	1 0.0%	0 0.0%	484 16.4%	3 0.1%	0 0.0%	99.2% 0.8%
5	0 0.0%	0 0.0%	0 0.0%	3 0.1%	529 18.0%	0 0.0%	99.4% 0.6%
6	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	537 18.2%	99.8% 0.2%
	99.8% 0.2%	96.0% 4.0%	99.8% 0.2%	98.6% 1.4%	99.4% 0.6%	100% 0.0%	98.9% 1.1%
	1	2	3	4	5	6	
	Target Class						

Fig. 11. Evaluating the proposed method using the confusion matrix for the testing dataset

The Receiver Operating Characteristic (ROC) curve is a graphical representation of a classifier's ability to distinguish between classes. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The Area Under the ROC Curve (AUC) is a single scalar value that summarizes the overall performance of the model; a higher AUC indicates better performance.

Figure 12 and Figure 13 show the ROC curves for the training and testing datasets. Each curve illustrates the trade-off between sensitivity and specificity across different thresholds.

For the training dataset (Figure 12), the ROC curve is positioned at the top-left corner, indicating a high true positive rate and a zero false positive rate. The AUC for the training set is 1, demonstrating perfect model performance on the training data.

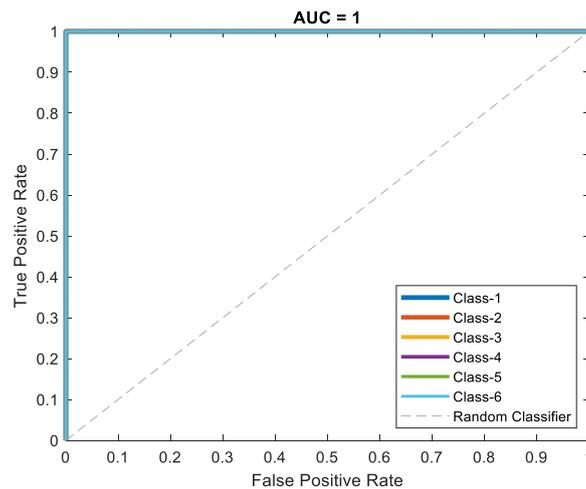


Fig. 12. Evaluating the proposed method using the ROC curve for the training dataset

Similarly, the ROC curve for the testing dataset (Figure 13) maintains a favourable position, though slightly less optimal than the training curve. The AUC for the testing set is 0.99989, reflecting the model's excellent performance on unseen data, which is typical and indicates good generalization.

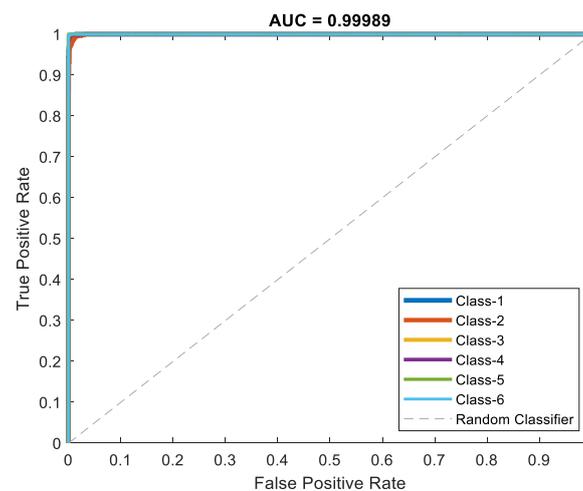


Fig. 13. Evaluating the proposed method using the ROC curve for the testing dataset

To quantitatively evaluate the model's performance, we use several metrics of accuracy, precision, recall and F1 score, as mentioned in Section IV. These metrics provide a comprehensive understanding of the model's effectiveness from different perspectives. Accuracy measures the overall correctness, precision indicates the correctness of positive predictions, recall measures the ability to identify all positive instances and the F1 score balances precision and recall.

Figure 14 and Figure 15 present bar charts illustrating these metrics for the training and testing datasets.

The bar chart for the training dataset (Figure 14) shows 100% values across all metrics, indicating that the model performs perfectly in correctly predicting the classes.

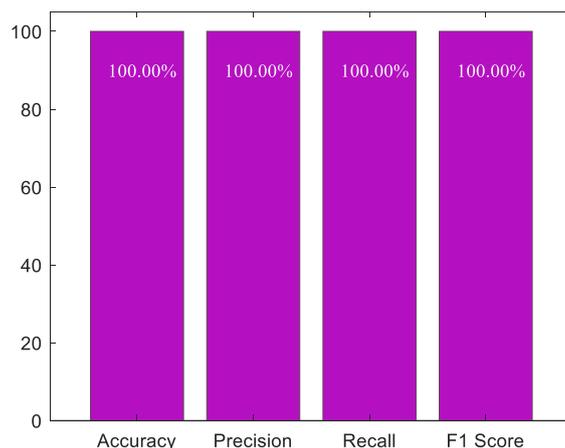


Fig. 14. Evaluating the proposed method using the evaluation metrics for the training dataset

For the testing dataset (Figure 15), the metrics are slightly lower than those of the training set but still demonstrate strong performance, with all values above 98.9%. This slight decrease is normal and suggests that the model maintains good predictive power when applied to new data.

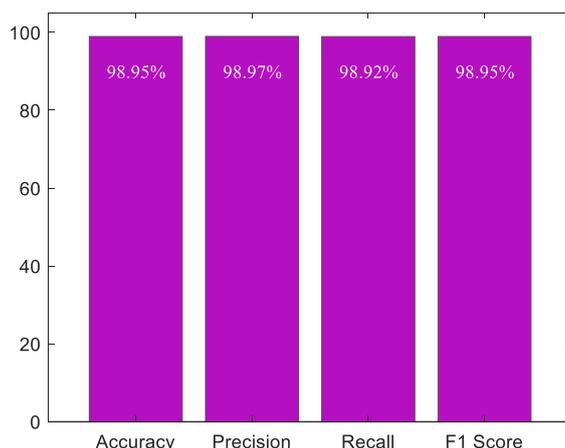


Fig. 15. Evaluating the proposed method using the evaluation metrics for the testing dataset

4.2 Comparison

This section presents an extensive comparison between the proposed method and other approaches found in the literature. First, each of these methods will be described, followed by a concise tabular overview for easy comparison.

A deep neural network that combines convolutional layers with LSTM was proposed by Xia *et al.*, [41], enabling the automatic extraction of activity features and classification with few model parameters. The specific contribution of this method lies in its ability to process temporal sequences effectively, utilizing a combination of LSTM for capturing time-dependent patterns and convolutional layers for spatial feature extraction. This architecture allows the model to achieve high accuracy with reduced complexity, making it suitable for real-time applications. LSTM, a variant of the recurrent neural network. Evaluation of model performance was conducted on three public datasets (UCI, WISDM and OPPORTUNITY). In comparison to the proposed method, which may incorporate

different techniques or architectures, the advantages of this LSTM-CNN approach include automatic feature extraction and fewer model parameters. However, potential disadvantages could include a reliance on large datasets for training and a possible limitation in generalization to unseen data types. Finally, the overall accuracy of the model was assessed, yielding 95.78% in the UCI-HAR dataset, 95.85% in the WISDM dataset and 92.63% in the OPPORTUNITY dataset.

According to Challa *et al.*, [42], a robust classification model for Human Activity Recognition using wearable sensor data is designed by employing a hybrid approach combining CNN and BiLSTM. The proposed multibranch CNN-BiLSTM network conducts automatic feature extraction from raw sensor data with minimal preprocessing. The specific contribution of this method lies in its ability to learn both local features through the convolutional layers and long-term dependencies through the BiLSTM layers, making it well-suited for time-series data inherent in HAR tasks. Utilizing CNN and BiLSTM enables the model to learn both local features and long-term dependencies in sequential data. This approach relates to the proposed method by demonstrating a different architecture that combines both CNN and BiLSTM, potentially offering complementary strengths to the proposed technique. The model's performance is evaluated using three benchmark datasets: WISDM, UCI-HAR and PAMAP2, resulting in accuracies of 96.05%, 96.37% and 94.29% respectively. Advantages of this hybrid method include enhanced feature extraction capabilities and improved performance on diverse activities, while potential disadvantages might involve increased model complexity and computational requirements compared to simpler models.

A model based on a Deep Neural Network, incorporating a Convolutional Neural Network and a Gated Recurrent Unit, is proposed by Dua *et al.*, [43] for end-to-end activity classification with automatic feature extraction. The specific contribution of this method is its ability to perform both feature extraction and classification within a single model architecture, minimizing the need for manual feature engineering, which is often time-consuming and requires domain expertise. Experiments in this study utilize raw data from wearable sensors with minimal preprocessing, omitting handcrafted feature extraction techniques. This approach relates to the proposed method by demonstrating an alternative deep learning architecture that emphasizes end-to-end learning and the automation of feature extraction, potentially enhancing generalization and performance. Accuracy rates achieved on the UCI-HAR, WISDM and PAMAP2 datasets are 96.20%, 97.21% and 95.27% respectively. The advantages of this method include the elimination of manual feature extraction, improved classification performance on diverse datasets and the ability to capture both local and long-term dependencies through the combined use of CNN and GRU layers. However, potential disadvantages may involve increased model complexity and computational demands compared to more straightforward architectures.

In a study by Mekruksavanich *et al.*, [44], a novel framework is introduced for multi-class wearable user identification, focusing on the recognition of human behaviour using deep learning models. The specific contribution of this method lies in its approach to utilizing deep learning for biometric user identification through the analysis of human activity, leveraging sensory data from tri-axial gyroscopes and accelerometers of wearable devices. Sensory data from tri-axial gyroscopes and accelerometers of wearable devices are utilized to gather comprehensive user information during various activities. This method relates to the proposed method by demonstrating an effective way to harness deep learning for user identification, which could complement existing activity recognition systems. Additionally, a series of experiments were conducted to validate this work, demonstrating the effectiveness of the proposed framework. Results from the two fundamental models, CNN and LSTM deep learning revealed that the highest accuracy achieved for all users was 91.77% and 92.43%, respectively, on UCI HAR and USC HAD datasets. The advantages of this method include its ability to automatically extract relevant features from wearable sensor data, thus minimizing the need for

manual feature engineering and potentially enhancing user identification accuracy. However, disadvantages may involve limitations in generalization to other contexts or datasets and reliance on specific sensor configurations that may not be universally available across all wearable devices.

All mentioned methods obtain appropriate performance for intrusion detection. However, the proposed method with a great ability for avoiding overfitting, outperforms all of them. Table 2 reports the summary of the comparative study.

Table 2

The comparison of the proposed method with other SDN intrusion detection methods

Reference	Method	Dataset	Accuracy
[41]	Deep neural network with LSTM and convolutional layers	UCI-HAR	95.78%
[42]	Hybrid CNN-BiLSTM	UCI-HAR	96.37%
[43]	Deep Neural Network with CNN and Gated Recurrent Unit (GRU)	UCI-HAR	96.20%
[44]	Convolutional Neural Network Long Short-Term Memory	UCI HAR	91.77%
The proposed method	PCA dimensionality reduction and LSTM optimized with GWO	UCI HAR	98.95

As recommended for future studies, machine learning methods and some algorithms, such as the GA-PRM algorithm's adaptability and potential for efficient navigation in healthcare and automation [45,46].

5. Conclusions

This study introduced an innovative human activity recognition (HAR) method that combines Principal Component Analysis (PCA) for dimensionality reduction with Long Short-Term Memory (LSTM) networks optimized using the Grey Wolf Optimization (GWO) algorithm. The integration of these advanced techniques aimed to tackle significant challenges in HAR, such as high dimensionality and the complexity of hyperparameter optimization, which are crucial for developing robust and efficient recognition systems.

The initial phase of our method employed PCA to reduce the input feature space from 561 to 196 features. This reduction preserved more than 99% of the original data variance, ensuring that the most informative aspects of the data were retained while discarding redundant or irrelevant features. By doing so, we enhanced computational efficiency and mitigated the risk of overfitting, which is a common issue when dealing with high-dimensional datasets. The PCA step effectively transformed the feature space, making the subsequent modelling process more manageable and focused.

Following dimensionality reduction, the LSTM network's hyperparameters were optimized using the GWO algorithm. This optimization process was critical in fine-tuning the network to capture the complex temporal dependencies present in human activity data. The hyperparameters tuned included the number of hidden units, learning rate, learning rate drop frequency and batch size. By leveraging GWO's ability to search for optimal hyperparameters, we ensured that the LSTM network was not only effectively trained but also capable of achieving high performance.

The proposed method was rigorously tested and demonstrated a remarkable accuracy of 98.95%. This outstanding performance underscores the robustness and efficacy of our approach in human activity recognition tasks. The combination of PCA and GWO proved to be a powerful strategy, enhancing both the efficiency and accuracy of the LSTM network. The success of our method highlights its potential for real-world applications, where accurate and efficient HAR is essential.

Moreover, the study addresses several critical challenges in the HAR domain. The use of PCA significantly reduced the feature space, addressing the issue of high dimensionality that often leads to overfitting and increased computational burden. The GWO optimization of LSTM hyperparameters tackled the challenge of model tuning, ensuring that the network parameters were set to values that maximize learning efficiency and accuracy.

The integration of PCA and GWO not only improved the model's performance during training but also enhanced its generalization capability. This is particularly important for real-world applications, where the model needs to perform reliably on new, unseen data.

Acknowledgement

This research was not funded by any grant.

References

- [1] Yusup, Norfadzlan, Izzatul Nabila Sarbini, Dayang Nurfatimah Awang Iskandar, Azlan Mohd Zain and Didik Dwi Prasetya. "Enhancing Wearable-Based Human Activity Recognition with Binary Nature-Inspired Optimization Algorithms for Feature Selection." *Journal of Advanced Research in Applied Sciences and Engineering Technology* (2024): 113-124.
- [2] Mohamed, Raihani, Nur Hidayah Azizan, Thinagaran Perumal, Syaifulnizam Abd Manaf, Erzam Marlisah and Medria Kusuma Dewi Hardhienata. "Discovering and Recognizing of Imbalance Human Activity in Healthcare Monitoring using Data Resampling Technique and Decision Tree Model." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 33, no. 2 (2023): 340-350. <https://doi.org/10.37934/araset.33.2.340350>
- [3] Liftawi, Sana M., Iyad A. Yousef and Mahmoud W. Kaied. "An Accurate Classification for Human Sport Activity Recognition from Sensor Data Using Hybrid Machine Learning Models." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 52, no. 1 (2025): 68-88. <https://doi.org/10.37934/araset.52.1.6888>
- [4] Dentamaro, Vincenzo, Vincenzo Gattulli, Donato Impedovo and Fabio Manca. "Human activity recognition with smartphone-integrated sensors: A survey." *Expert Systems with Applications* (2024): 123143. <https://doi.org/10.1016/j.eswa.2024.123143>
- [5] Sabeeh, Sarah. "Enhancing Robotic Grasping Performance through Data-Driven Analysis." *Misan Journal of Engineering Sciences* 3, no. 1 (2024): 134-156. <https://doi.org/10.61263/mjes.v3i1.78>
- [6] Wu, Jia, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei and Si-Hao Deng. "Hyperparameter optimization for machine learning models based on Bayesian optimization." *Journal of Electronic Science and Technology* 17, no. 1 (2019): 26-40.
- [7] Ankalaki, Shilpa and M. N. Thippeswamy. "Optimized Convolutional Neural Network Using Hierarchical Particle Swarm Optimization for Sensor Based Human Activity Recognition." *SN Computer Science* 5, no. 5 (2024): 447. <https://doi.org/10.1007/s42979-024-02794-5>
- [8] Wan, Shaohua, Lianyong Qi, Xiaolong Xu, Chao Tong and Zonghua Gu. "Deep learning models for real-time human activity recognition with smartphones." *mobile networks and applications* 25, no. 2 (2020): 743-755. <https://doi.org/10.1007/s11036-019-01445-x>
- [9] Mutegeki, Ronald and Dong Seog Han. "A CNN-LSTM approach to human activity recognition." In *2020 international conference on artificial intelligence in information and communication (ICAIIIC)*, pp. 362-366. IEEE, 2020. <https://doi.org/10.1109/ICAIIIC48513.2020.9065078>
- [10] Zhou, Xiaokang, Wei Liang, I. Kevin, Kai Wang, Hao Wang, Laurence T. Yang and Qun Jin. "Deep-learning-enhanced human activity recognition for Internet of healthcare things." *IEEE Internet of Things Journal* 7, no. 7 (2020): 6429-6438. <https://doi.org/10.1109/JIOT.2020.2985082>
- [11] Dirgová Luptáková, Iveta, Martin Kubovčík and Jiří Pospíchal. "Wearable sensor-based human activity recognition with transformer model." *Sensors* 22, no. 5 (2022): 1911. <https://doi.org/10.3390/s22051911>
- [12] Ahmed, Nadeem, Jahir Ibna Rafiq and Md Rashedul Islam. "Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model." *Sensors* 20, no. 1 (2020): 317. <https://doi.org/10.3390/s20010317>
- [13] Mekruksavanich, Sakorn and Anuchit Jitpattanukul. "Lstm networks using smartphone data for sensor-based human activity recognition in smart homes." *Sensors* 21, no. 5 (2021): 1636. <https://doi.org/10.3390/s21051636>
- [14] Dahou, Abdelghani, Mohammed AA Al-qaness, Mohamed Abd Elaziz and Ahmed Helmi. "Human activity recognition in IoT applications using arithmetic optimization algorithm and deep learning." *Measurement* 199 (2022): 111445. <https://doi.org/10.1016/j.measurement.2022.111445>

- [15] Helmi, Ahmed M., Mohammed AA Al-qaness, Abdelghani Dahou and Mohamed Abd Elaziz. "Human activity recognition using marine predators algorithm with deep learning." *Future Generation Computer Systems* 142 (2023): 340-350. <https://doi.org/10.1016/j.future.2023.01.006>
- [16] Xiao, Zhiwen, Xin Xu, Huanlai Xing, Fuhong Song, Xinhan Wang and Bowen Zhao. "A federated learning system with enhanced feature extraction for human activity recognition." *Knowledge-Based Systems* 229 (2021): 107338. <https://doi.org/10.1016/j.knosys.2021.107338>
- [17] Gil-Martín, Manuel, Rubén San-Segundo, Fernando Fernández-Martínez and Ricardo de Córdoba. "Human activity recognition adapted to the type of movement." *Computers & Electrical Engineering* 88 (2020): 106822. <https://doi.org/10.1016/j.compeleceng.2020.106822>
- [18] Mim, Taima Rahman, Maliha Amatullah, Sadia Afreen, Mohammad Abu Yousuf, Shahadat Uddin, Salem A. Alyami, Khondokar Fida Hasan and Mohammad Ali Moni. "GRU-INC: An inception-attention based approach using GRU for human activity recognition." *Expert Systems with Applications* 216 (2023): 119419. <https://doi.org/10.1016/j.eswa.2022.119419>
- [19] Priyadarshini, Ishaani, Rohit Sharma, Dhowmya Bhatt and M. Al-Numay. "Human activity recognition in cyber-physical systems using optimized machine learning techniques." *Cluster Computing* 26, no. 4 (2023): 2199-2215. <https://doi.org/10.1007/s10586-022-03662-8>
- [20] Abdel-Basset, Mohamed, Hossam Hawash, Victor Chang, Ripon K. Chakraborty and Michael Ryan. "Deep learning for heterogeneous human activity recognition in complex iot applications." *IEEE Internet of Things Journal* 9, no. 8 (2020): 5653-5665. <https://doi.org/10.1109/JIOT.2020.3038416>
- [21] Iloga, Sylvain, Alexandre Bordat, Julien Le Kerneec and Olivier Romain. "Human activity recognition based on acceleration data from smartphones using HMMs." *IEEE Access* 9 (2021): 139336-139351. <https://doi.org/10.1109/ACCESS.2021.3117336>
- [22] Bijalwan, Vishwanath, Vijay Bhaskar Semwal and Vishal Gupta. "Wearable sensor-based pattern mining for human activity recognition: Deep learning approach." *Industrial Robot: the international journal of robotics research and application* 49, no. 1 (2022): 21-33. <https://doi.org/10.1108/IR-09-2020-0187>
- [23] Lawal, Isah A. and Sophia Bano. "Deep human activity recognition with localisation of wearable sensors." *IEEE Access* 8 (2020): 155060-155070. <https://doi.org/10.1109/ACCESS.2020.3017681>
- [24] Chen, Jingcheng, Yining Sun and Shaoming Sun. "Improving human activity recognition performance by data fusion and feature engineering." *Sensors* 21, no. 3 (2021): 692. <https://doi.org/10.3390/s21030692>
- [25] Haresamudram, Harish, Irfan Essa and Thomas Plötz. "Contrastive predictive coding for human activity recognition." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, no. 2 (2021): 1-26. <https://doi.org/10.1145/3463506>
- [26] Zhu, Simin, Ronny Gerhard Guendel, Alexander Yarovoy and Francesco Fioranelli. "Continuous human activity recognition with distributed radar sensor networks and CNN-RNN architectures." *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022): 1-15. <https://doi.org/10.1109/TGRS.2022.3189746>
- [27] Tong, Lina, Hanghang Ma, Qianzhi Lin, Jiaji He and Liang Peng. "A novel deep learning Bi-GRU-I model for real-time human activity recognition using inertial sensors." *IEEE Sensors Journal* 22, no. 6 (2022): 6164-6174. <https://doi.org/10.1109/JSEN.2022.3148431>
- [28] Shavit, Yoli and Itzik Klein. "Boosting inertial-based human activity recognition with transformers." *IEEE Access* 9 (2021): 53540-53547. <https://doi.org/10.1109/ACCESS.2021.3070646>
- [29] Li, Chenglin, Di Niu, Bei Jiang, Xiao Zuo and Jianming Yang. "Meta-har: Federated representation learning for human activity recognition." In *Proceedings of the web conference 2021*, pp. 912-922. 2021. <https://doi.org/10.1145/3442381.3450006>
- [30] Ferrari, Anna, Daniela Micucci, Marco Mobilio and Paolo Napolitano. "Deep learning and model personalization in sensor-based human activity recognition." *Journal of Reliable Intelligent Environments* 9, no. 1 (2023): 27-39. <https://doi.org/10.1007/s40860-021-00167-w>
- [31] Ihianle, Isibor Kennedy, Augustine O. Nwajana, Solomon Henry Ebebuwa, Richard I. Otuka, Kayode Owa and Mobolaji O. Orisatoki. "A deep learning approach for human activities recognition from multimodal sensing devices." *IEEE Access* 8 (2020): 179028-179038. <https://doi.org/10.1109/ACCESS.2020.3027979>
- [32] Park, Hyunseo, Nakyoung Kim, Gyeong Ho Lee and Jun Kyun Choi. "MultiCNN-FilterLSTM: Resource-efficient sensor-based human activity recognition in IoT applications." *Future Generation Computer Systems* 139 (2023): 196-209. <https://doi.org/10.1016/j.future.2022.09.024>
- [33] Greenacre, Michael, Patrick JF Groenen, Trevor Hastie, Alfonso Iodice d'Enza, Angelos Markos and Elena Tuzhilina. "Principal component analysis." *Nature Reviews Methods Primers* 2, no. 1 (2022): 100. <https://doi.org/10.1038/s43586-022-00184-w>
- [34] Siami-Namini, Sima, Neda Tavakoli and Akbar Siami Namin. "A comparative analysis of forecasting financial time series using arima, lstm and bilstm." *arXiv preprint arXiv:1911.09512* (2019).

- [35] Chen, Weifeng, Fei Zheng, Shanping Gao and Kai Hu. "An LSTM with differential structure and its application in action recognition." *Mathematical Problems in Engineering* 2022, no. 1 (2022): 7316396. <https://doi.org/10.1155/2022/7316396>
- [36] Asghari, Kayvan, Mohammad Masdari, Farhad Soleimanian Gharehchopogh and Rahim Saneifard. "A chaotic and hybrid gray wolf-whale algorithm for solving continuous optimization problems." *Progress in Artificial Intelligence* 10, no. 3 (2021): 349-374. <https://doi.org/10.1007/s13748-021-00244-4>
- [37] Dai, Shuyu, Dongxiao Niu and Yan Li. "Daily peak load forecasting based on complete ensemble empirical mode decomposition with adaptive noise and support vector machine optimized by modified grey wolf optimization algorithm." *Energies* 11, no. 1 (2018): 163. <https://doi.org/10.3390/en11010163>
- [38] Li, Yu, Xiaoxiao Lin and Jingsen Liu. "An improved gray wolf optimization algorithm to solve engineering problems." *Sustainability* 13, no. 6 (2021): 3208. <https://doi.org/10.3390/su13063208>
- [39] Tonkal, Özgür, Hüseyin Polat, Erdal Başaran, Zafer Cömert and Ramazan Kocaoğlu. "Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking." *Electronics* 10, no. 11 (2021): 1227. <https://doi.org/10.3390/electronics10111227>
- [40] Anguita, Davide, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge Luis Reyes-Ortiz. "A public domain dataset for human activity recognition using smartphones." In *Esann*, vol. 3, p. 3. 2013.
- [41] Xia, Kun, Jianguang Huang and Hanyu Wang. "LSTM-CNN architecture for human activity recognition." *IEEE Access* 8 (2020): 56855-56866. <https://doi.org/10.1109/ACCESS.2020.2982225>
- [42] Challa, Sravan Kumar, Akhilesh Kumar and Vijay Bhaskar Semwal. "A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data." *The Visual Computer* 38, no. 12 (2022): 4095-4109. <https://doi.org/10.1007/s00371-021-02283-3>
- [43] Dua, Nidhi, Shiva Nand Singh and Vijay Bhaskar Semwal. "Multi-input CNN-GRU based human activity recognition using wearable sensors." *Computing* 103, no. 7 (2021): 1461-1478. <https://doi.org/10.1007/s00607-021-00928-8>
- [44] Mekruksavanich, Sakorn and Anuchit Jitpattanukul. "Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models." *Electronics* 10, no. 3 (2021): 308. <https://doi.org/10.3390/electronics10030308>
- [45] Sabeeh, Sarah and Israa S. Al-Furati. "Comparative Analysis of GA-PRM Algorithm Performance in Simulation and Real-World Robotics Applications." *Misan Journal of Engineering Sciences* 2, no. 2 (2023): 12-37. <https://doi.org/10.61263/mjes.v2i2.57>
- [46] Sabeeh, Sarah. "Forecasting Robot Movement with Sensor Readings and Multi-Layer Perceptron Models." *Misan Journal of Engineering Sciences* 3, no. 1 (2024): 63-83. <https://doi.org/10.61263/mjes.v3i1.77>