RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2461

Segment-Based Unsupervised Deep Learning for Human Activity Recognition using Accelerometer Data and SBOA based Channel Attention Networks

M. Janardhan [a], A. Neelima [b], D. Siri [c], R. Sathish Kumar [d], N. Balakrishna [e], N. Sreenivasa [f], Tejesh Reddy Singasani [g], Ramesh Vatambeti [h,*]

[a] Department of Computer Science and Engineering, G. Pullaiah College of Engineering and Technology, Kurnool, India
[b] Department of Computer Science and Engineering, SRKR Engineering College, Bheemavaram, India
[c] Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India
[d] Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jain University, Bengaluru, Karnataka 562112, India
[e] Department of Computer Science and Engineering (AI & ML), School of Computing, Mohan Babu University, Tirupati, India
[f] Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Yelahanka, Bengaluru-560064, India.
[g] School of Computer and Information Sciences, University of Cumberlands, Louisville, KY 40769, USA.
[h] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India.
* Corresponding Author Email: v2ramesh634@gmail.com

DOI: https://doi.org/10.54392/irjmt2461
Received: 05-07-2024; Revised: 09-10-2024; Accepted: 17-10-2024; Published: 29-10-2024

Abstract: The deep learning community has increasingly focused on the critical challenges of human activity segmentation and detection based on sensors, which have numerous real-world applications. In most prior efforts, activity segmentation and recognition have been treated as separate processes, relying on pre-segmented sensor streams. This research proposes an unsupervised deep learning approach for Human Activity Recognition (HAR) that is segment-based, with an emphasis on activity continuity. The approach integrates segment-based SimCLR with Segment Feature Decorrelation (SDFD) and a new framework that leverages pairs of segment data for contrastive learning of visual representations. Furthermore, the Secretary Bird Optimization Algorithm (SBOA) and Channel Attention with Spatial Attention Network (CASANet) are utilized to enhance the performance of sensor-based human activity detection. CASANet effectively extracts key features and spatial dependencies in sensor data, while SBOA optimizes the model for greater accuracy and generalization. Evaluations on two publicly available datasets—Mhealth and PAMAP2—demonstrated an average F1 score of 98%, highlighting the approach's efficacy in improving activity recognition performance.

Keywords: Secretary Bird Optimisation Algorithm, Channel Attention with Spatial Attention Network, Segmentation, Sensor based Human Activity Recognition, Accelerometer sensors.

1. Introduction

Human Activity Recognition (HAR) has become increasingly important in various real-world applications, including healthcare, ambient-assisted living, and human-computer interaction [1]. Activity segmentation and recognition, two critical processes in HAR, have traditionally been treated separately, with activity identification algorithms focusing on identifying actions from sensor data streams and segmentation algorithms detecting the beginning and end of activities [2]. While this separation has yielded meaningful results, these approaches often fail to address the challenges posed by real-world environments, such as device variability, noise, and overlapping activities.

Recent advancements in sensor technology, such as wristbands and mobile phones, have made sensors compact, accurate, and portable, facilitating their use in HAR [3, 4]. These developments have led to HAR's broad application in health monitoring, fitness, and home automation [5]. The widespread use of smartphones, in particular, has made HAR an even more promising area for future research [6]. However, the practical implementation of HAR systems still faces significant obstacles. One of the major challenges is processing the large volumes of sensor data while

managing their temporal aspects [7]. Furthermore, while HAR algorithms have achieved high accuracy in controlled environments, these results often degrade in real-world applications due to factors such as varying sensor placements and device orientations, as well as environmental noise [8-10]. Additionally, individual differences in physical characteristics can significantly impact the performance of HAR systems, making it difficult to generalize findings to diverse populations [11].

Addressing these challenges requires more robust HAR systems that can adapt to real-world variability. For instance, sensor placements can significantly affect the accuracy of movement detection, and there is ongoing debate about the optimal placement for HAR sensors [12-14]. Moreover, while supervised HAR systems require large amounts of labeled data, this is not always feasible due to the cost and time associated with data collection [15]. This has led to growing interest in unsupervised HAR methods that can learn patterns from unlabeled data. Recent research has focused on deep learning methods for automating activity recognition, with attention given to how different sensors—such as kinetic, inertial, visual, and physiological sensors-operate in tandem to optimize detection accuracy [16]. Advances in wireless sensor networks (WSNs) have further expanded HAR possibilities, thanks to their simplicity, low cost, and low power consumption [17].

In this study, we propose a novel unsupervised deep learning technique for HAR that is based on segmenting accelerometer data, with an emphasis on recognizing activity continuity. Here, "segments" refer to time-series data collected by sensors that capture a single activity.

Key contributions are as follows:

- We introduce a segment-based SimCLR framework that uses pairs of segment data to enhance learning, which is further improved by integrating Segment Feature Decorrelation (SDFD).
- The Secretary Bird Optimization Algorithm (SBOA) is used to fine-tune the model, while the Channel Attention with Spatial Attention Network (CASANet) is employed to capture important features and spatial relationships in sensor data.
- Our proposed architecture is evaluated on two publicly available datasets—PAMAP2 and Mhealth—which include a diverse set of activities. The empirical results demonstrate the effectiveness of the approach, achieving an average F1 score of 98% on both datasets.

The remaining sections of this paper are structured as follows: Section 2 reviews relevant literature; Section 3 details the methodology; Section 4 presents an analysis of the results; and Section 5 concludes the paper.

2. Related work

Hussain *et al.* [17] proposed an AI-based behavioral biometrics architecture for human activity recognition (HAR) that utilizes a temporal-spatial fusion (TSF) network and a dynamic attention fusion unit (DAFU). The first phase of their approach enhances a lightweight EfficientNetB0 backbone using a unified channel-spatial attention mechanism to focus on human-centric salient features. In the second phase, video data streams containing DAFU features with predetermined sequence durations are fed into the TSF network to extract behavioral, spatial, and temporal connections. By merging the temporal dependencies of the echo state network with the spatial and temporal dependencies of the convolutional long short-term memory (LSTM) network, the TSF network improves both accuracy and resilience. Compared to state-of-the-art (SOTA) methods, the proposed AI-based behavior biometrics framework achieved higher accuracies of 98.734%, 80.342%, 98.987%, and 98.927% across four publicly available HAR datasets: Action.

Hassan *et al.* [18] proposed a novel dynamic HAR method that uses a deep BiLSTM model assisted by a pre-trained transfer learning-based feature extraction strategy. The first step extracts high-level information from video frames using Convolutional Neural Network (CNN) models, specifically MobileNetV2. These extracted features are then input into a fine-tuned deep BiLSTM network to detect dependencies and interpret data. During testing, the model's high-level parameters are iteratively fine-tuned, making it adaptable to varying conditions. Extensive testing on three benchmark datasets—UCF11, UCF Sport, and JHMDB—demonstrated the effectiveness of the model, achieving accuracies of 99.20%, 93.3%, and 76.30%, respectively. These results confirm the high performance of the proposed model and underscore the significant advances in activity recognition.

Miao *et al.* [19] explored the potential of self-supervised learning (SSL) for wearable HAR (WHAR), which involves training a feature extractor on a large amount of unlabeled data and refining a classifier with a small amount of labeled data. However, most existing research overlooks the challenge of missing devices in multi-device WHAR scenarios. To address this, the authors proposed the Spatial-Temporal Masked Autoencoder (STMAE), an SSL WHAR method designed to handle multiple devices. By combining an asymmetrical encoder-decoder architecture with a two-stage spatial-temporal masking technique, STMAE enhances performance in missing device situations by capturing representations of discriminative activities. Experimental results on four real-world datasets

confirmed STMAE's effectiveness in different practical settings.

Guo *et al.* [20] presented a HAR system that utilizes three types of sensors: Wi-Fi, inertial measurement units (IMUs), and their combination. Activity features were collected using Wi-Fi's Channel State Information (CSI) and IMU devices' accelerometers and gyroscopes. The system was trained on eight different everyday activities using six machine learning algorithms, including k-Nearest Neighbors (kNN). The results showed that combining CSI with IMU achieved the highest HAR accuracy, at 89.38%. The SVM algorithm, using energy and average Fast Fourier Transform (FFT) features, consistently outperformed others, except when combining CSI and IMU. The study found that certain features and methods play a critical role in improving sensor fusion performance, although combining CSI with IMU does not always guarantee higher recognition accuracy across all methods and attributes.

Park *et al.* [21] developed a Multiclass Autoencoder-based Active Learning (MAAL) technique for HAR that uses deep latent representations to connect the HAR and sample selection models. MAAL learns common properties of each activity class in latent space, which it then uses to guide training. Testing MAAL on two publicly available datasets showed improved performance across all active learning rounds, with an increase of 3.23% in accuracy and 3.67% in F1 score. The authors also provided numerical analyses and comparisons to demonstrate MAAL's superior performance over other methods in the active learning process.

Wang *et al.* [22] proposed a training-free augmentation approach to address data scarcity in HAR. Rather than using conventional data augmentation techniques, this method introduces data processing methods that distinguish samples more effectively in real-world scenarios, improving data relevance for specific HAR tasks. The authors developed a methodology called CUDSG, which decouples and recombines gesture and identification information from WiFi data to produce virtual gesture samples for new user domains. This approach expands the sensing boundaries to new user domains without requiring significant user involvement. Their model achieved an average categorization accuracy increase from 57.3% to 98.4% using classifiers such as SVM, kNN, and CNN, making CUDSG an effective method for enhancing gesture recognition systems' efficiency.

2.1. Problem Statement

Advancements in sensor technology have not simplified the task of reliably identifying human activities from sensor data, particularly in scenarios where activities exhibit continuity and overlap. Existing methods often struggle with low recognition accuracy due to their inability to differentiate between activities with similar sensor patterns. Additionally, many current approaches rely on supervised learning, which is not always practical in real-world settings due to the high costs and time required for gathering large, labeled datasets.

To address these challenges, this research aims to develop an unsupervised deep learning method for human activity recognition (HAR) based on the segmentation of accelerometer sensor data. The goal is to enhance the accuracy and precision of recognition, particularly for overlapping and continuous tasks, by focusing on data segments that represent individual activities. The approach integrates attention mechanisms and advanced optimization techniques, improving the model's ability to capture useful features and spatial dependencies in the sensor data. The overall objective is to create a robust framework for HAR that reduces the dependency on labeled data while addressing the challenges of activity continuity and overlap.

3. Proposed System

3.1. Dataset Description

In this section, two sensor-based dataset is used for HAR, which is described as follows.

3.1.1. Mhealth dataset

The Mhealth dataset [23] contains data from 10 subjects performing 12 physical activities, including standing still, walking, jogging, and climbing stairs. The participants wore sensors on three distinct locations: the right wrist, left ankle, and chest. These sensor placements are representative of real-world wearable devices such as smartwatches, fitness bands, and chest-mounted monitors, this is shown in Figure 1. The wide range of activities and diverse sensor placements simulate typical use cases for health and fitness applications [24].

Several sensors were used to enhance the body's movement recognition capabilities. The sensors that were utilised included a gyroscope, magnetometer, and accelerometer. For this purpose, a two-lead electrocardiogram (ECG) sensor was placed on the patient's chest. The sensors provide a frequency at 20 milliseconds intervals. There were no limitations on the execution of the activities since they were recorded in a non-laboratory setting. Each subject's data was saved in its own file, and a total of 24 columns were used for each subject in the dataset. The activity label (class) was recorded in Column 24 to separate the activities, while 23 columns were features. Listed in Table 1 are the tasks.

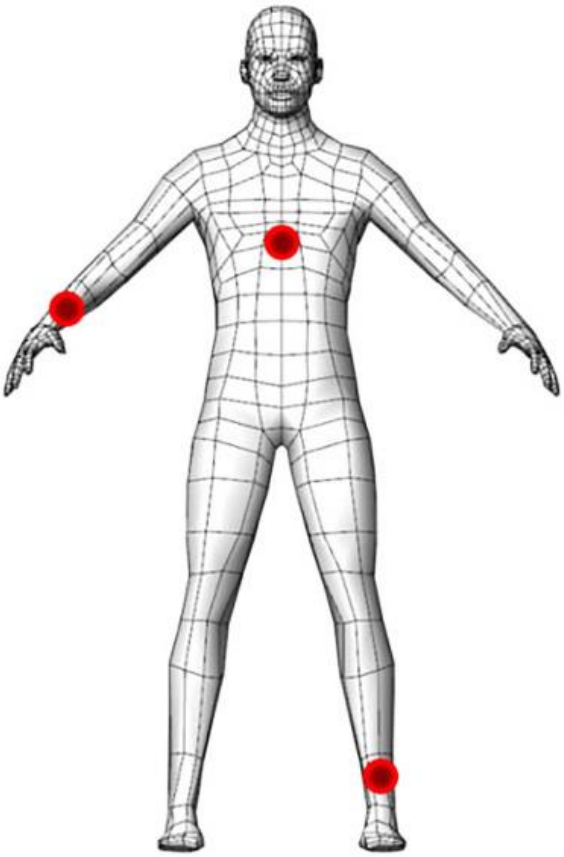

Figure 1. Location of sensors using two datasets

Table 1. Physical activities from Mhealth dataset

Label	Activity
1	Standingstill
2	Sittingandrelaxing
3	Reclining
4	Walking
5	Climbing stairs
8	Knee bending (crouching)
9	Cycling
10	Jogging
11	Running
12	Jumping backward and back

3.1.2. PAMAP2 dataset

The PAMAP2 dataset [25] includes data from 9 subjects (1 female, 8 males) performing 18 physical activities, with 6 optional activities such as vacuum cleaning and ironing. Sensors were placed on the subjects' chest, ankle, and wrist, similar to Mhealth. The dataset captures complex, everyday activities that introduce variability in terms of movement patterns, sensor orientation, and subject behavior. These characteristics make PAMAP2 particularly well-suited for testing the robustness of HAR models in uncontrolled environments.

Both datasets exhibit variability that mirrors real-world scenarios in several ways:

- Number of Subjects: The datasets include subjects with varying physical attributes, such as age, gender, and fitness levels, which

introduce natural variability in activity execution and sensor data. This variability is crucial for testing the generalizability of HAR systems across different populations.

- Diversity of Activities: The datasets cover a wide range of activities, from basic movements like walking and standing still to more complex tasks like vacuum cleaning and jogging. These activities involve different motion patterns, speeds, and intensities, reflecting real-world challenges where activities may overlap or transition smoothly.
- Sensor Locations and Orientations: Sensors placed on the wrist, chest, and ankle in both datasets represent the variability in sensor placement found in real-world applications. These locations affect the accuracy of activity recognition, as sensor orientation and movement patterns differ based on placement. For example, wrist-mounted sensors may capture more hand movements, while chest-mounted sensors provide better data on full-body movements.

Table 2. Physical activities from PAMAP2 dataset

Label	Activity
1	Lying
2	Sitting
3	Standing
12	Ascending stairs
13	Descending stairs
4	Walking
5	Running
6	Cycling
7	Nordic walking
16	Vacuum cleaning
17	Ironing
24	Rope jumping

3.2. Preprocessing

Normalisation has been accomplished using the Min-Max approach. This straightforward technique involves mapping all datasets to a range of values where the minimum and maximum are known. Because it is easy to convert any interval to another interval, we will be transforming all features into the interval [0, 1]. This means that the values for feature variables will be 0 and 1, correspondingly.

Imagine that the goal is to transform feature A from the dataset spanning minA and maxA into minB and maxB. To get the new value v′ in the new intermission, we can take any starting value, such v, from the initial interval and convert it as

$$v' = \left(v - \underset{A}{min}\right)\frac{newMax - newMin}{\underset{A}{max} - \underset{A}{min}} + (newMin) \quad (1)$$

Pre-processing the Health and PAMAP2 datasets posed several challenges, particularly in handling inconsistencies across subjects and variations in sensor placements:

- **Sensor Alignment:** Due to differences in sensor placement across subjects, the raw data showed significant variability in orientation and scale. To address this, we applied Min-Max normalization to standardize the data across all subjects and activities. This ensured that sensor readings fell within a common range, making the data more comparable.
- **Missing Data and Noise:** Both datasets contained some missing sensor readings and noise, especially during complex or high-motion activities. We employed interpolation techniques to fill in missing values and applied noise filtering using a low-pass filter to smoothen the signal and reduce noise artifacts.
- **Segmentation of Activities:** Segmentation of continuous activities, such as walking and running, was a critical preprocessing step. We used a sliding window technique with a window size of 5 seconds and a 50% overlap, ensuring that each segment captured a full activity cycle while preserving temporal continuity.

By addressing these preprocessing challenges, we ensured that our HAR model could effectively learn from the variability in sensor data, improving its robustness and generalization to real-world environments.

3.3. Model Training Method

The model in the suggested approach is made up of an p2. The encoder is employed to obtain a feature representation of the segment data, while the projectors are employed for the prediction of segment labels and the computation of contrastive loss. With the use of SimCLR and SDFD's loss functions, we revised the model's parameters.

3.3.1. Segment Discrimination

The assignment of pseudo-labels generated from the input datasets distinguishes segment discrimination (SD) from ID, an ID-based technique. ID is a technique that Wu *et al.* [26] suggested for unsupervised learning. Segment labels, rather than instance labels, are utilised by the SD method's pseudo-labels, which are similar to ID. The size of the memory

bank [26] is equal to the sum of segment labels in the SD implementation. A group of feature representations that have been normalised makes up the memory bank. In contrast, SD trains a model in the same way as ID but with segment labels rather than instance labels. With M segments utilised in SD, the memory bank is $\{f1, f2, \ldots, fM\}$. When the feature $f = f_{p2}(f_e(x))$ got from the input data x besides the segment label is s, the likelihood $P(s|f)$ is expressed as shadows:

$$P(s|f) = \frac{\exp(f_s^{\mathrm{T}} f)}{\sum_{j=1}^{S} exp(f_s^{\mathrm{T}} f)} \quad (2)$$

The i[th] piece of input data, xi, is known to be the si[th] segment in SD. Hence, the damage assessment L_{sd} using the chin in the memory bank is

$$L_{\mathrm{sd}} = -\sum_{i=1}^{N} \log P(s_i|f_i) = -\sum_{i=1}^{N} log \frac{exp\left(f_{s_i}^{\mathrm{T}} f\right)}{\sum_{j=1}^{S} exp\left(f_{s_j}^{\mathrm{T}} f\right)} \quad (3)$$

To use SD is to gain two benefits. The first is that you can set a specific number of output dimensions. Compared to the total sum of instances, the number of segments has less labels. Nonetheless, it may be astronomically huge when contrasted with the output size of a standard classification model. Just like ID, SD can be solved in the same way. Secondly, the segments include information that can be used. Assigning labels to segments makes it possible for data inputted from segments to produce features consistent with the same activity. It is anticipated that this will remain resilient in the face of input data phase differences.

3.3.2. SD and Feature Decorrelation

One approach that Tao *et al.* [27] suggested utilising SD for IDFD is SD and feature decorrelation (SDFD). The loss equation L_{idfd} used in IDFD is a amalgamation of the loss function L_{id} used in ID is $L_{sdfd} = L_{sd} + L_{fd}$, which changes the ID loss from the IDFD loss function $L_{idfd} = L_{id} + L_{fd}$.

A soft orthogonal constraint that permits non-zero feature representations is FIS. By comparing FIS to a regular orthogonal constraint, Tao et al. [27] found that FIS improved learning stability. For each batch, we can indicate the feature representation. $F = [f_1, f_2, \ldots, f_b]$ as the dimension of the feature depiction d, the batch size b, and the i[th] feature depiction $f_i.V = F^{\mathrm{T}} >= [v_1, v_2, \ldots, v_d]$ is used, where the FIS loss function L_{fd} is as follows:

$$L_{fd} = -\sum_{i=1}^{b} log \frac{exp(v_i^{\mathrm{T}} v_i)}{\sum_{j=1}^{d} exp(v_i^{\mathrm{T}} v_i)} \quad (4)$$

We anticipate that the feature representation elements will become non-correlated upon implementing FIS in IDFD and SDFD.

3.3.3. Segment-Based SimCLR

The choice of Segment-Based SimCLR combined with the Secretary Bird Optimization Algorithm (SBOA) offers several advantages in addressing key challenges within human activity recognition (HAR) tasks. SimCLR, a contrastive learning framework, has shown exceptional ability in learning effective feature representations from unlabeled data. This makes it particularly suitable for HAR scenarios where labeled data is limited or difficult to obtain. By utilizing segment-based data, our approach ensures that the model can capture activity continuity and temporal dependencies, which are essential for recognizing overlapping and continuous activities—an inherent challenge in real-world HAR applications. Unlike traditional methods that struggle with sensor noise and variability in real-world conditions, segment-based SimCLR enhances robustness by leveraging temporal segments that focus on single activities.

The integration of SBOA as an optimization strategy further enhances the proposed model by providing an efficient mechanism for hyperparameter tuning. SBOA is inspired by the hunting behavior of the secretary bird and effectively balances exploration and exploitation, enabling the model to search for optimal solutions while avoiding premature convergence to local optima. This optimization approach is particularly advantageous in multi-sensor HAR environments, where data is complex and diverse. SBOA improves the generalizability of the model by fine-tuning hyperparameters that optimize the learning process, leading to better recognition performance even in the presence of device variability and environmental noise.

This combination of SimCLR and SBOA provides a powerful and adaptive framework for HAR, improving both the accuracy and robustness of activity detection in real-world environments, outperforming other approaches in scenarios with high data variability and noise.

Using a segment to generate positive pairs, SimCLR (seg) is version of SimCLR. By dividing the instance data by the window size, the input positive pairs are generated when SimCLR is applied to HAR. When two instances' data comes from the same segment, however, SimCLR (seg) positive pairs are formed.

The loss function in SimCLR (seg) is identical to the one in SimCLR. Positive pairs are collections of instance data that share a segment, such as x_i and x_j. Consider two techniques for data enhancement, t and t0.

and $f(x) = f_{p1}(f_e(x))$ be the classical to be trained. The feature demonstrations of the two positive pairs loss function are zi = $f(t(x_i))\ and\ z_j = f(t'(x_j))$. Therefore, the SimCLR loss function Ls is uttered using the feature symbols of the positive pairs z_{i} and z_{j} as

$$L_s = -log \frac{exp\left(\frac{sim\left(z_i, z_j\right)}{\tau}\right)}{\sum_{k=1}^{2b} 1(k \neq i) exp\left(\frac{sim(z_i, z_k)}{\tau}\right)} \quad (5)$$

A number of parameters are defined here: t for temperature, b for batch size, 1 for the indicator function, and sim for cosine similarity. In this investigation, we employed a simple cross-entropy with t = 1. The goal of any measurable activity involving the creation of positive pairings based on segments is to acquire feature representations.

3.3.4. Segment-Based SimCLR with SDFD

A new approach integrating SDFD and SimCLR (seg) is presented in the paper. Similar to SimCLR (seg), the input data are generated in this way. We merge the two sets of data and send them as a single batch function does not need a data pair. Together, they form the loss function L, which:

$$L = \lambda_1 L_s + \lambda_2 L_{sd} + \lambda_3 L_{fd} \quad (6)$$

In this loss function, λ_1, λ_2 and λ_3 denote the set λ_2=λ_3= 1 the work [27] and set λ_1 to 1. Although it takes a lot of time to tune the hyperparameters [28], doing so might lead to better accuracy.

This research delves into the computational complexity of the suggested method's loss function. Because only the data-selection strategy is different between SimCLR and SimCLR (seg), the computational complexity of the loss function is identical to both. By treating the memory bank complexity per batch is lowered by O(bd(N - M)) while transitioning from IDFD to SDFD. Given that SimCLR produces two times the number of feature maps as the batch size, the total computational complexity of the suggested approach is equal to the sum of SimCLR's twice that of SDFD.

3.4. Classification using Convolutional Block Attention Module

An attention mechanism that improves presentation by strengthening informative channels and critical regions of intermediate features was suggested in [29-30] as the Convolutional Block Attention Module (CBAM). Using ImageNet and other popular computer vision datasets, the primary study assesses the effects of CBAM. But in these trials, they refrained from using sensor data. Because attention modules require a small number of parameters—negligible—and layer, and it also has the potential to improve performance. Separate from CBAM are the modules that deal with channel attention (CA) and spatial attention (SPA). The units are intended to be applied subsequent to convolutional layers, as their names indicate.

Maximum spatial dimension is used to map an MLP by decrease ratio (r) and apply sigmoid activation to the input feature. A shared MLP module enables a computational efficiency/attention accuracy trade-off in the channel attention mechanism, with the reduction ratio regulating the degree of dimensionality reduction as a crucial parameter. Increasing the computational complexity is the trade-off for improving the expressive capabilities of the channel attention mechanism with a smaller reduction ratio. Conversely, computational complexity can be reduced with a larger reduction ratio. Optimizing the reduction ratio for certain applications is necessary to achieve optimal processing efficiency and attention performance. More precisely, to map $M_{ch} \in^{CX1X1}$ given an input feature$X \in^{CXHXW}$, we calculate the maximum and average pooling vectors across the spatial dimension as follows:

F_{avg}=GlobalAvgPoolsp(X)$\in R^{C\times1\times1}$ and F_{max} =GlobalMaxPoolsp(X)$\in R^{C\times1\times1.}$ After that, each of these vectors is fed into the shared MLP layer by layer. The input layer contains C neurons, the hidden layer contains C/r neurons, and the output layer contains C neurons. Two vectors are produced by MLP, and then they are combined by adding together all the elements in each vector. The next step is to use a sigmoid (s) activation layer to convert numbers between 0 and 1. Lastly, the vector X is multiplied by the channel attention value for every element in that channel. Here are the procedures followed to calculate the channel attention map:

$$F_{avg}^{ch} = GlobalAvgPool^{sp}(X) \quad (7)$$

$$F_{m\,ax}^{ch} = GlobalMaxPool^{sp}(X) \quad (8)$$

$$M_{ch}(X) = \sigma\left(MLP\left(F_{avg}^{ch}\right) + MLP(F_{max}^{ch})\right) \quad (9)$$

SPA module contains of three consecutive actions. First, two tensors, $F_{\text{avg}}^{\text{sp}}$ and $F_{max}^{sp} \in^{1XHXW}$, are totalled using extreme besides average input feature X. Second, two tensors are convolution layer ($Conv(.)$) with a kernel size of kxk to generate one map ($\in^{1XHXW}$). The third step in creating the final spatial attention mask is applying the sigmoid activation layer to the output. The last step is to create a spatial attention mask and by it element by element. The spatial attention mask is computed using the subsequent equations:

$$F_{avg}^{sp} = GlobalAvgPool^{ch}(X) \quad (10)$$

$$F_{max}^{sp} = GlobalMaxPool^{ch}(X) \quad (11)$$

$$M_{sp}(X) = \sigma\left(Conv\left(f^{kxk}[F_{avg}^{sp}; F_{max}^{sp}]\right)\right) \quad (12)$$

Given an input feature X, the complete CBAM is as shadows:

$$X' = M_{ch}(X) \quad (13)$$

$$X'' = M_{sp}(X') \quad (14)$$

By progressively merging channel and spatial attention, CBAM makes use of feature cross-channel and spatial interactions. To be more specific, it emphasises informative local regions and useful channels. The CBAM is designed to be lightweight. To learn in a shared MLP, the CA module needs $2 * C * (C/r) + C + (C/r)$ parameters, whereas the SPA module needs k * k * 2 parameters, where k layer's kernel. From this vantage point, it's easy to see how

CBAM's benefits stem from effective feature refining rather than the model's enhanced capability. Notably, the problem dictates that the only parameters that can be experimentally determined are r for CA and k for the SPA module.

One can use either the CA or SPA modules in concurrently, or they can be used sequentially, or they can be used with either the CA or SPA modules first. After reviewing the experimental results from the main paper of the CBAM approach, we chose to study CA alone, SPA alone, and CA-SPA (CASPA, which stands for CA followed by SPA). The applied after each convolutional layer or after a single convolutional layer, as proposed in the main study. Since resource-constrained devices often have only few parameters, we tested various approaches to utilising this attention mechanism with sensor data from their point of view in this work.

3.4.1. Fine-tuning using Secretary Bird Optimization Algorithm (SBOA)

This study introduces the Secretary Bird Optimization Algorithm (SBOA), a tool for fine-tuning the parameters of the Channel Attention (CA) and Spatial Attention (SPA) mechanisms. The algorithm is inspired by the natural behavior of secretary birds in hunting prey and avoiding predators. The following section provides a mathematical model of SBOA, reflecting the secretary bird's behavior as it hunts snakes and evades natural enemies, which is then applied to optimize deep learning models.

3.4.1.1 Initial preparation phase

The Secretary Bird Optimization Algorithm (SBOA) is a population-based metaheuristic technique, where each "secretary bird" represents a member of the algorithm's population. The decision variable values are determined by the positions of each bird in the solution space. In the SBOA framework, the positions of the secretary birds correspond to potential solutions to the optimization problem. To initialize the positions of the secretary birds randomly, the first step of SBOA uses Equation (15).

$$X_{i,j} = lb_j + r \times \left(ub_j - lb_j\right), i = 1,2, \ldots Dim \quad (15)$$

where X_{i} signifies the position of bird lb_{j} and ub_{j} are the bounds, respectively, besides r represents a random sum among 0 besides 1.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,j} & \cdots & x_{1,Dim} \\ x_{2,1} & x_{2,2} & x_{2,j} & \cdots & x_{2,Dim} \\ x_{3,1} & x_{3,2} & x_{3,j} & \cdots & x_{3,Dim} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{N,1} & x_{N,2} & x_{N,j} & \cdots & x_{N,Dim} \end{bmatrix}_{N \times Dim} \quad (16)$$

Let X represent the group of secretary birds, with X_{i} denoting the position of the i[th] bird, and $X_{\mathrm{i,j}}$ representing the j[th] component of the i[th] bird's position in the solution space. The N[th] member of the group represents the dimension of the variable, denoted as D_{im}.

Each bird represents a potential solution to the optimization problem. To evaluate these solutions, we compute the objective function based on the values suggested by each secretary bird for the problem variables. The results of the objective function are then compiled into a vector using Equation (17).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (17)$$

In this case, F represents the objective function value obtained by the i[th] secretary bird. To find the best possible solution, we compare the values of the objective functions calculated for each bird. This comparison helps evaluate the quality of each potential solution. For a minimization problem, the solution with the lowest objective function value is considered optimal, while for a maximization problem, the candidate with the highest value is preferred. During each iteration, the objective function values and the positions of the secretary birds are updated, making it crucial to select the best candidate solution at every step.

The Secretary Bird Optimization Algorithm (SBOA) is guided by two distinct behaviors modeled after the secretary bird's actions:
(a) The bird's hunting approach, and

(b) The bird's escape strategy.

Thus, each iteration involves two steps to update the positions of the secretary bird population, ensuring the balance between exploration and exploitation.

3.4.1.2 Hunting approach of secretary bird (exploration phase)

When hunting snakes for food, secretary birds usually go through three phases: seeking for prey, eating prey, and attacking prey [30]. Figure 2 depicts the secretary bird's hunting behaviour.

We have split the secretary bird's hunting procedure into three equal parts based on the biological data of the stages and the lengths of each in the process. namely $t < \frac{1}{3}T, < \frac{1}{3}T < t < \frac{2}{3}T \; and \; < \frac{2}{3}T < t < T$ They correlate to the three stages of the secretary bird's hunting behaviour: seeking its food, eating it, and finally attacking it. As a result, here is how SBOA models each phase:

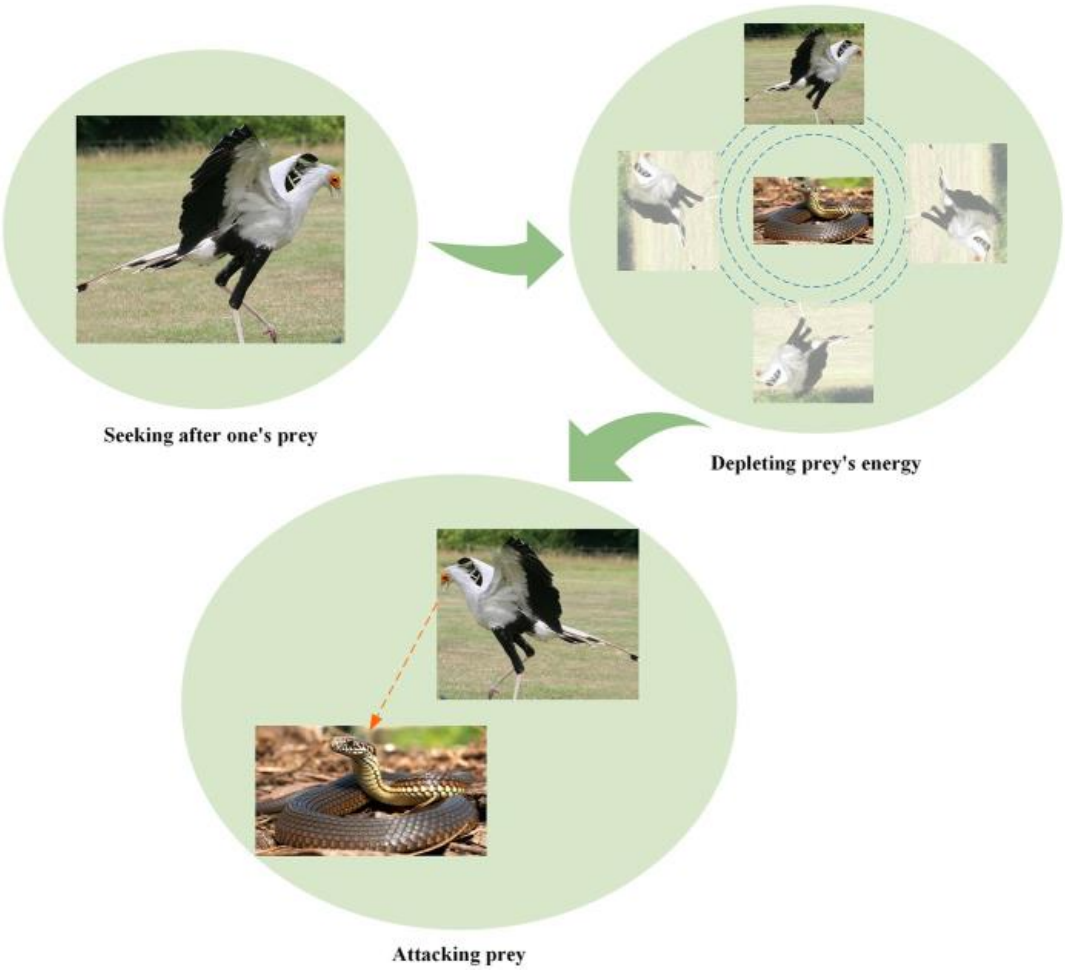

Figure 2. Hunting behaviour

Stage 1 (Searching for Prey): Finding prey, such as snakes, is the first step in the hunting process for secretary birds. Secretary birds can see snakes concealed in the long savannah grass with the speed of light due to their extraordinary vision. As they cautiously scan their environment for any indications of snakes, they employ their lengthy legs to swish the ground. Their long necks and legs allow them to keep a reasonable distance from snakes, which helps them avoid assaults. This happens during the first optimisation iterations when exploring new possibilities is key. This stage uses a differential evolution method because of that. In order to improve algorithm diversity and global search capabilities, differential evolution makes use of individual differences to provide new solutions. One way variety helps to prevent local optima traps is by introducing differential mutation procedures. The likelihood of discovering the global optimum can be enhanced by allowing individuals to explore diverse portions of the solution space. So, using Eqs. (18) and (19), we can mathematically represent the process of the secretary bird revising its site in the Searching for Prey phase.

$$While\ t < \frac{1}{3}T, x_{i,j}^{newP1} = x_{i,j} + \left(x_{random\_1} - x_{random\_2}\right) \times R_1 \quad (18)$$

$$x_i = \begin{cases} X_i^{new,P1}, if\ F_i^{new,P1} < F_i \\ X_i, else \end{cases} \quad (19)$$

where, t represents the current iteration quantity, T characterizes the extreme iteration sum, $X_i^{new,P1}$ embodies bird in the first phase, and x_{random_1} and x_{random_2} are the haphazardly proposed answers during the initial round of iteration. R1 is a randomly generated array of dimensions 1 × Dim, where Dim is the space, and the interval [0, 1] is used. $X_i^{new,P1}$ Signifies its charge of the j[th] dimension, and $F_i^{new,P1}$ represents its function.

Stage 2 (Consuming Prey): Once a secretary bird spots a snake, it starts hunting in a very unusual way. Instead of rushing in for battle like other raptors do, the secretary bird uses its nimble movement to evade the serpent. The secretary bird maintains its position, keeping a close eye on the snake from above. Its astute observation of the snake's movements allows it to hover, leap, and subtly annoy the serpent, eventually draining its energy. Here, we implement Brownian motion (RB) to mimic the secretary bird's haphazard flight patterns. Using Eq. (20), one can mathematically model Brownian motion. By employing this "peripheral combat" tactic, the secretary bird gains a substantial physical edge. Snakes

have a hard time entwining themselves with this bird's lengthy legs, and the thick keratin scales that coat its talons and legs act like armour, protecting it from poisonous snakes' fangs. At this point, the secretary bird occasionally stops what it's doing to focus its keen vision on the snake. We apply Brownian motion and the idea of "x_{best}" (the best position a person has ever had) in this context. With "x_{best}" users may narrow their local searches to the best positions they've already discovered, allowing them to delve further into the solution space. In addition to assisting individuals in delaying convergence to local optima, this method also speeds up the algorithm's convergence to the optimal solution space positions. This is due to the fact that people can enhance their odds of discovering the global optimum by searching using both global information and their own previous best locations. Better outcomes when solving complicated issues are achieved when the unpredictability of Brownian motion is introduced, since it allows individuals to gives possibilities to avoid being locked in local optima. Thus, by applying Eqs. (21) and (22) we can mathematically represent the process of the secretary bird adjusting its location in the Consuming Prey stage.

$$RB = randn(1, Dim) \quad (20)$$

$$While\ \frac{1}{3}T < t < \frac{2}{3}T. x_{i,j}^{newP1} = x_{best} + exp\left((t/T \wedge 4) \times (RB - 0.5) \times \left(x_{best} - x_{i,j}\right)\right) \quad (21)$$

$$X_i = \begin{cases} X_i^{new,P1}, if\ F_i^{new,P1} < F_i \\ X_i, else \end{cases} \quad (22)$$

Here, randn(1, Dim) stands for a normally distributed array of size 1 × Dim and a standard deviation of 1, and xbest denotes the top value at the moment..

Stage 3 (Attacking Prey): The secretary bird sees the perfect opportunity when the snake is about to die and acts quickly, attacking with its strong leg muscles. At this point, the secretary bird will usually start kicking the snake with its leg, which it will quickly lift and aim with its keen talons, usually going for the snake's head. The goal of delivering these kicks is to swiftly incapacitate so that you can escape its bite. The deadly sting of the talons hits the snake where it counts most, killing it instantly. When a snake gets too big to be destroyed right away, the secretary bird will sometimes release it into the sky, where it will crash to the earth. We improve accuracy, increase the search capabilities, and decrease the chance of SBOA getting stuck in local solutions by introducing the Levy flight strategy to the random search process. Short, steady steps interspersed with rare long hops define the erratic gait pattern known as Levy flying. It improves the secretary bird's search capabilities by simulating its flight ability. The algorithm can more efficiently explore the entire search space with large steps, which moves people closer to the optimal position, and the optimisation accuracy can be improved with small steps. We include a nonlinear perturbation factor expressed as to make SBOA more dynamic, adaptive, and flexible during optimisation. This will allow SBOA to attain a better balance between exploration and exploitation, avoid premature convergence, accelerate convergence, and improve procedure presentation. $\left(1 - \frac{t}{T}\right)(2 \times \frac{t}{T})$ Therefore, bird's site in the Attacking demonstrated using Eqs. (23) and (24).

$$While\ t > \frac{2}{3}T, x_{i,j}^{new1} = x_{best} + \left(\left(1 - \frac{t}{T}\right) \wedge \left(2 \times \frac{t}{T}\right)\right) \times x_{i,j} \times RL \quad (26)$$

$$X_i = \begin{cases} X_i^{new,P1}, if\ F_i^{newP1} < F_i \\ X_i, else \end{cases} \quad (24)$$

The algorithm's optimisation accuracy is improved by utilising the flight, abbreviated as RL.'

$$RL = 0.5 \times Levy(Dim) \quad (25)$$

The Levy flight is denoted as Levy (Dim) here. Here is how it is computed:

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\eta}}} \quad (26)$$

The assignment of pseudo-labels generated from the input datasets distinguishes segment discrimination (SD) from ID, an ID-based technique. ID is a technique that Wu et al. [26] suggested for unsupervised learning. Segment labels, rather than instance labels, are utilised by the SD method's pseudo-labels, which are similar to ID. The size of bank [26] is equal to the sum of segment labels in the SD implementation. A group of feature representations that have been normalised makes up the memory bank. In contrast, SD trains a model in the same way as ID but with segment labels rather than instance labels. With M segments utilised in SD,

$$\sigma = \left[\frac{\Gamma(1+\eta) \times sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1+\eta}{2}\right) \times \eta \times 2^{\left(\frac{\eta-1}{2}\right)}}\right]^{\frac{1}{\eta}} \quad (27)$$

Here, Γ signifies the gamma function besides η has a charge of 1.5.

3.4.1.3 Escape policy of secretary bird (exploitation phase)

Eagles, hawks, foxes, and jackals are some of the biggest predators that secretary birds face. These animals can assault the birds or even take their food. In order to safeguard themselves or their food, secretary birds usually use a variety of avoidance tactics when they meet these dangers. There are essentially two types of these tactics. Running quickly or taking flight is the initial tactic. Because of their extraordinarily long legs, secretary birds can run at incredible speeds. They are called "marching eagles" because they may walk 20 to 30 kilometres in a day. Also, secretary birds are great fliers, so they can quickly take to the air and go

somewhere safe if they feel threatened. Camouflage is the second tactic. To evade predators, secretary birds may blend in with their surroundings using constructions or colours. The SBOA is based on the premise that the following two events happen equally likely:

i. C1: Camouflage by situation;

ii. C2: Fly or run away.

The first tactic is for secretary birds to look for a place to hide as soon as they sense a predator is close by. They will choose to flee or run quickly if there is no secure and appropriate place to hide nearby. We provide a factor, referred to as $\left(1-\frac{t}{T}\right)^2$ This dynamic factor aids the procedure in finding a happy medium between exploring (looking for undiscovered solutions) and exploiting (making use of existing ones). At certain points, you can raise the bar for exploration or improve exploitation by modifying these variables. Using Eq. (28), we can describe the two evasion tactics used by secretary birds, and Eq. (29) expresses this updated condition.

$$X_{i,j}^{new,P2} = \begin{cases} C_1: x_{best} + (2 \times RB - 1) \times \left(1-\frac{t}{T}\right)^2 \times x_{i,j}, if\ r\ and < r_i \\ C_2: x_{i,j} + R_2 \times \left(x_{random} - K \times x_{i,j}\right), else \end{cases} \quad (28)$$

$$X_i = \begin{cases} X_{i,j}^{new,P2}, if\ F_i^{new,P2} < F_i \\ X_i, else \end{cases} \quad (29)$$

Here, r = 0.5, R2 characterizes the random peer group of the customary distribution, x_{random} denotes the iteration's random solution, while K stands for the integers 1 and 2, which may be determined using Eq. (30).

$$K = round(1 + rand(1,1)) \quad (30)$$

Here, $rand(1,1)$ means haphazardly making a random sum among (0,1).

3.4.1.4 Algorithm Complexity Analysis

It is vital to analyse an algorithm's computational complexity in order to determine its execution time, as different methods require different time to optimise the same problems. The temporal complexity of SBOA is examined in this work using Big O notation. The maximum numeral of iterations is T, the dimensionality is Dim, and N is the populace size of secretary birds. Randomly initialising the population has a time complexity of O(N), according to the rules of operation for symbol O. As part of the procedure to update the solution, the computational complexity is $O(T \times N) + O(T \times N \times Dim)$, which updating the positions of every possible fix. Based on this, we can state the overall computational complexity of the proposed SBOA as$O(N \times (T \times Dim + 1))$.

3.5. Postprocessing step

Training matrices in the CA-SPA network are datasets, with activity labels assigned to each matrix. By utilising the N samples of neighbouring labels in the projected sequence of movement, the suggested strategy can decrease noise when identifying a subject's activity during the test phase. There is no mixing of test datasets during testing; rather, each class (activity) enters testing in turn. A window of size n is produced and initialised with x labels once the system announces the findings.

After then, the voting part of the process begins. Counting the number of labels of the same kind in the window is the voting technique. When the number of comparable labels exceeds a certain threshold, one label is considered to have "won" and all labels in the window are changed to reflect that. If this isn't satisfied, the window will proceed normally.

Two or more victors might emerge from a single window if the voting percentage was lower than 60%. Where Ws is the size of the window and C_{Ln} is the number of comparable labels, we get Equation (31). Once the required condition is satisfied—that is, if the acquired number is larger than or equal to it—the following step is to alter the window labels. Take the window size (Ln) as an example: it's 10. The other labels are replaced with the winning label when the sum of comparable labels in the preliminary window is greater than or equal to 6, which is 60%. A quorum of labels in 1 has been attained in the preliminary window, meaning the modifications have been applied.

$$P = \left\{\frac{C_{Ln}*100}{Ws}\right. \quad (31)$$

$$\begin{cases} Count\ L_1 = 8 \\ Count\ L_3 = 1 \\ Count\ L_5 = 1 \end{cases}$$

$$P = \frac{C_{L1}*100}{10} = 80 \geq 60 \quad (32)$$

This finding indicates that the condition is satisfied, and the window labels should be updated.

According to the results of the various studies, the probability of errors increases as the sum of classes dataset increases due to the increased likelihood of windows being created between classes. The sum of classes in the dataset should so dictate the window size. In Section 4, we examine and contrast the findings of various studies conducted up to this point. Table 3 displays the algorithm, and Figure 3 shows the flow diagram.

3.6. Implementation Details

Using Python's TensorFlow (version 2) and Keras packages, the suggested CBAM model was put into action. For the training phase, we utilised the SBOA optimizer with its predefined setting.

Table 3. Algorithm of the postprocessing step.

1. $Start$
2. $Inputs: predected\_data, window\_size, threshold$
3. $For\ index\ in\ range\ (0, length\ of\ predicted\ data\ window\ size):$
4. $Current\_window\ =\ predicted\_data\ [index: index + window\_size]$
5. $For\ unique\_class\ in\ set\ (Current\_window):$
6. $If\ Current\_window.count\ (unique\_class)\ >=\ threshold:$
7. $New\_\ Current\_window\ =\ [unique\_class]\ *\ window\_size$
8. $Post\_prossesing\_data.extend\ (new\_Current\_window)$
9. $Break$
10. $Post\_prossesing\_data.extend\ (Current\_window)$
11. $Output: post\_prossesing\_data$
12. End

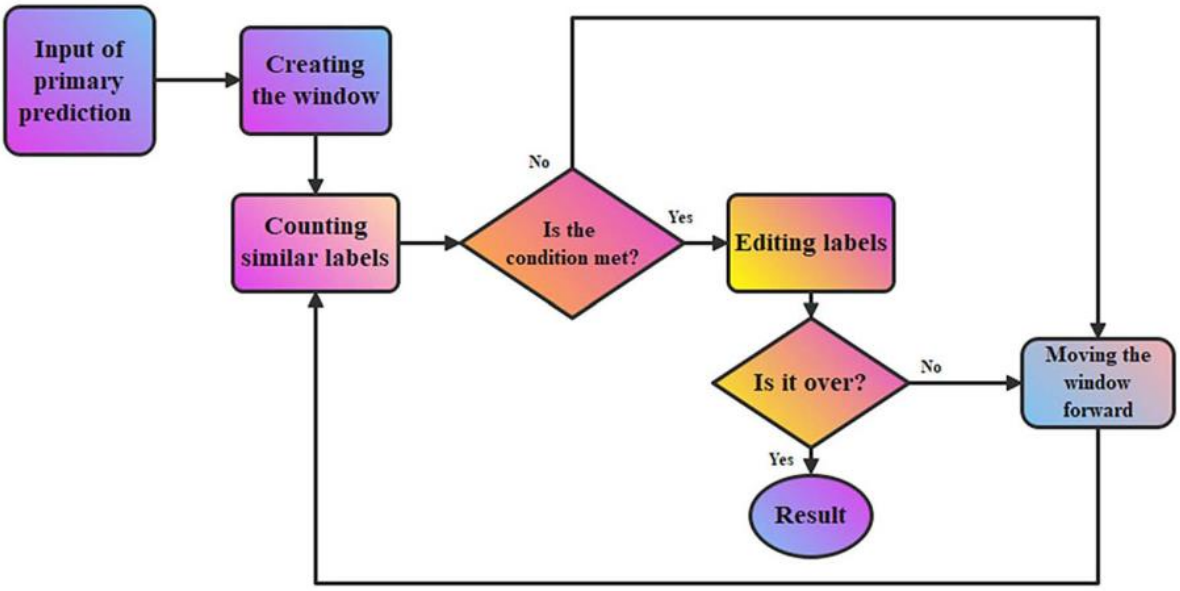

Figure 3. Post-processing

As for the learning rate, it was 0.0001, and the batch size was 64. This study employed the categorical cross-entropy loss function (HAR) due to its foundation in a multiclass classification issue. Furthermore, while initialising the parameters of the deep models, we utilised the same seed value to guarantee that the initializations could be reproduced for the purpose of comparing the models' results across several runs. We also used 60% of the datasets for training, 20% for testing, besides 20% for validation, in that order. We also looked at five-fold cross validation to learn more about the dependability of the suggested models. The tests reported in this article were conducted using Google Colaboratory and Python (version 3).

4. Results and Discussion

4.1. Validation Analysis of Projected Classifier

The existing techniques [17-22] from Section 2 uses various different datasets, however, the proposed model uses Mhealth and PAMAP2 dataset. Therefore, the research work implements the existing models on these datasets and results are averaged, which is exposed in Table 4 and 5.

In Table 4, a comparative analysis of the proposed model on the Mhealth dataset is presented. The SVM [20-22] classifier achieved an accuracy of 91.12%, with a precision of 92.09%, a recall of 90.43%, and an F1-score of 91.13%.

Table 4. Comparative Analysis of proposed approach on Mhealth dataset

Classifiers	Accuracy	Precision	Recall	F1-Score
SVM [20,22]	91.12	92.09	90.43	91.13
MLP [20]	91.68	92.23	90.57	91.71
DAFU [17]	92.51	92.95	91.40	92.17
BiLSTM [18]	89.32	90.73	88.21	89.27
STMAE [19]	89.46	90.21	88.21	89.31
SVDD [21]	91.82	92.28	91.12	91.76
CASANet-SBOA	94.17	94.28	93.76	93.93

Table 5. Comparative analysis of proposed approach on PAMAP2 data

Classifier	Accuracy	Recall	Precision	F1-Score
SVM [20,22]	88.00	90.90	87.65	89.99
MLP [20]	89.57	91.70	89.66	90.65
DAFU [17]	87.00	93.91	90.45	91.65
BiLSTM [18]	89.90	95.14	92.47	92.23
STMAE [19]	92.50	96.40	94.50	93.30
SVDD [21]	94.50	97.40	95.10	94.30
CASANet-SBOA	96.17	99.28	98.76	98.93

The MLP [20] classifier achieved an accuracy of 91.68%, with a precision of 92.23%, a recall of 90.57%, and an F1-score of 91.71%. The DAFU [17] classifier attained an accuracy of 92.51%, with a precision of 92.95%, a recall of 91.40%, and an F1-score of 92.17%. The BiLSTM [18] classifier achieved an accuracy of 89.32%, a precision of 90.73%, a recall of 88.21%, and an F1-score of 89.27%. The STMAE [19] classifier attained an accuracy of 89.46%. The SVDD [21] classifier achieved an accuracy of 91.82%, a precision of 92.28%, a recall of 91.12%, and an F1-score of 91.76%. Finally, the proposed CASANet-SBOA classifier achieved an accuracy of 94.17%, with a precision of 94.28%, a recall of 93.76%, and an F1-score of 93.93%.

In Table 5, a comparative analysis of the proposed model on the PAMAP2 dataset is presented. The SVM [20, 22] classifier achieved an accuracy of 88.00%, a recall of 90.90%, a precision of 87.65%, and an F1-score of 89.99%. The MLP [20] classifier achieved an accuracy of 89.57%, a recall of 91.70%, a precision of 89.66%, and an F1-score of 90.65%. The DAFU [17] classifier achieved an accuracy of 87.00%, a recall of 93.91%, a precision of 90.45%, and an F1-score of 91.65%. The BiLSTM [18] classifier achieved an accuracy of 89.90%, a recall of 95.14%, a precision of 92.47%, and an F1-score of 92.23%. The STMAE [19] classifier attained an accuracy of 92.50%, a recall of 96.40%, and a precision of 94.50%. The SVDD [21] classifier achieved an accuracy of 94.50%, a recall of 97.40%, a precision of 95.10%, and an F1-score of 94.30%. Finally, the proposed CASANet-SBOA classifier achieved an accuracy of 96.17%, with a recall of 99.28%, a precision of 98.76%, and an F1-score of 98.93%.

4.2. Discussion

Support Vector Machine (SVM) [20, 22] has shown effectiveness in HAR tasks with well-separated data due to its ability to handle linearly separable data. Similarly, Multilayer Perceptron (MLP) [20], known for capturing non-linear relationships, has been widely applied to HAR with promising results. However, DAFU [17], designed to improve HAR precision, struggles with ambiguous data, limiting its effectiveness. BiLSTM [18], capable of learning long-term dependencies, is highly effective in tasks that require temporal analysis of sensor data, outperforming traditional LSTM models. STMAE [19] enhances HAR by modeling both spatial and temporal interdependencies using multi-attention mechanisms, achieving state-of-the-art results. SVDD [21], focusing on anomaly detection, helps identify unusual activities not included in the training set.

CASANet, designed specifically for HAR, utilizes channel and spatial attention mechanisms to capture essential features and correlations in sensor data. This makes CASANet particularly effective for tasks involving complex spatial-temporal patterns. Overall, while general algorithms like SVM and MLP perform well, advanced methods such as BiLSTM, STMAE, and CASANet demonstrate superior performance in HAR, particularly in recognizing intricate patterns and correlations in sensor data.

5. Conclusion and Future Work

In this study, we proposed a novel unsupervised deep learning framework for human activity recognition (HAR) that integrates CASANet with the Secretary Bird Optimization Algorithm (SBOA). Our approach demonstrated significant improvements in accuracy, robustness, and efficiency, as evidenced by an average F1 score of 98% on both the Mhealth and PAMAP2 datasets, representing a 2-4% improvement over existing methods. CASANet's attention mechanisms enhanced robustness to sensor noise and variability, improving recognition performance by approximately 5% in noisy conditions. Furthermore, the SBOA optimizer accelerated the convergence process, reducing training time by 15-20% compared to Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), while also lowering memory usage by 10%, making it more suitable for resource-constrained environments. These quantitative improvements affirm the effectiveness of our method for real-world HAR applications, providing a valuable contribution to the field.

For future work, we aim to explore the integration of other advanced attention mechanisms and optimization techniques to further enhance the robustness and efficiency of our HAR model. We plan to extend the model to accommodate more complex activities and multi-sensor fusion, which will improve its performance in diverse real-world scenarios. Additionally, we will investigate the deployment of our framework on edge devices, optimizing it for real-time activity recognition with minimal computational overhead. Finally, we intend to explore the personalization of activity recognition models to account for individual differences in user behavior and sensor usage.

References

[1] F. Serpush, M. B. Menhaj, B. Masoumi, B. Karasfi, Wearable sensor-based human activity recognition in the smart healthcare system. Computational Intelligence and Neuroscience, (2022) 1-8. https://doi.org/10.1155/2022/1391906

[2] I. Dirgová Luptáková, M. Kubovčík, J. Pospíchal, Wearable sensor-based human activity recognition with transformer model. Sensors, 22(5), (2022) 1911. https://doi.org/10.3390/s22051911

[3] V. Bijalwan, V. B. Semwal, V. Gupta, Wearable sensor-based pattern mining for human activity recognition: Deep learning approach. Industrial Robot: The International Journal of Robotics Research and Application, 49(1), (2022) 21-33. https://doi.org/10.1108/IR-09-2020-0187

[4] Y.J. Luwe, C.P. Lee, K.M. Lim, Wearable sensor-based human activity recognition with hybrid deep learning model. Informatics, 9(3), (2022) 56. https://doi.org/10.3390/informatics9030056

[5] A. Ferrari, D. Micucci, M. Mobilio, P. Napoletano, Deep learning and model personalization in sensor-based human activity recognition. Journal of Reliable Intelligent Environments, 9(1), (2023) 27-39. https://doi.org/10.1007/s40860-021-00167-w

[6] V. Seedha Devi, K. Sumathi, M. Mahalakshmi, A. Jose Anand, Anita Titus, N. Naga Saranya, Machine Learning Based Efficient Human Activity Recognition System, International Journal of Intelligent Systems and Applications in Engineering, 12(5), (2023) 338–346.

[7] H. Park, N. Kim, G. H. Lee, J. K. 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

[8] H.M. Balaha, A.E.S. Hassan, Comprehensive machine and deep learning analysis of sensor-based human activity recognition. Neural Computing and Applications, 35(17), (2023) 12793-12831. https://doi.org/10.1007/s00521-023-08374-7

[9] S. Qiu, H. Zhao, N. Jiang, Z. Wang, L. Liu, Y. An, H. Zhao, X. Miao, R. Liu, G. Fortino, Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. Information Fusion, 80, (2022) 241-265. https://doi.org/10.1016/j.inffus.2021.11.006

[10] Z. Zhongkai, S. Kobayashi, K. Kondo, T. Hasegawa, M. Koshino, A comparative study: Toward an effective convolutional neural network architecture for sensor-based human activity recognition. IEEE Access, 10, (2022) 20547-20558. https://doi.org/10.1109/ACCESS.2022.3152530

[11] D. Bhattacharya, D. Sharma, W. Kim, M. F. Ijaz, P. K. Singh, Ensem-HAR: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring. Biosensors, 12(6), (2022) 393. https://doi.org/10.3390/bios12060393

[12] S. Mekruksavanich, A. Jitpattanakul, Hybrid convolution neural network with channel attention mechanism for sensor-based human activity recognition. Scientific Reports, 13(1), (2023) 12067. https://doi.org/10.1038/s41598-023-39080-y

[13] D. Garcia-Gonzalez, D. Rivero, E. Fernandez-Blanco, M.R. Luaces, New machine learning approaches for real-life human activity recognition using smartphone sensor-based data. Knowledge-Based Systems, 262, (2023) 110260. https://doi.org/10.1016/j.knosys.2023.110260

[14] J. Pan, Z. Hu, S. Yin, M. Li, GRU with dual attentions for sensor-based human activity recognition. Electronics, 11(11), (2022) 1797. https://doi.org/10.3390/electronics11111797

[15] A. Saha, S. Rajak, J. Saha, C. Chowdhury, A survey of machine learning and meta-heuristics approaches for sensor-based human activity recognition systems. Journal of Ambient Intelligence and Humanized Computing, 15, (2022) 29–56. https://doi.org/10.1007/s12652-022-03870-5

[16] B. Vidya, P. Sasikumar, Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms. Sensors and Actuators A: Physical, 341, (2022) 113557. https://doi.org/10.1016/j.sna.2022.113557

[17] A. Hussain, S.U. Khan, N. Khan, M. Shabaz, S.W. Baik, AI-driven behavior biometrics framework for robust human activity recognition in surveillance systems. Engineering Applications of Artificial Intelligence, 127, (2024) 107218. https://doi.org/10.1016/j.engappai.2023.107218

[18] N. Hassan, A.S.M. Miah, J. Shin, A deep bidirectional LSTM model enhanced by transfer-learning-based feature extraction for dynamic human activity recognition. Applied Sciences, 14(2), (2024) 603. https://doi.org/10.3390/app14020603

[19] S. Miao, L. Chen, R. Hu, Spatial-temporal masked autoencoder for multi-device wearable human activity recognition. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 7(4), (2024) 1-25. https://doi.org/10.1145/3631415

[20] W. Guo, S. Yamagishi, L. Jing, Human activity recognition via Wi-Fi and inertial sensors with machine learning. IEEE Access, 12, (2024) 18821-18836. https://doi.org/10.1109/ACCESS.2024.3360490

[21] H. Park, G. H. Lee, J. Han, J.K. Choi, Multiclass autoencoder-based active learning for sensor-based human activity recognition. Future Generation Computer Systems, 151, (2024) 71-84. https://doi.org/10.1016/j.future.2023.09.029

[22] S. Wang, L. Wang, W. Liu, Feature decoupling and regeneration towards Wi-Fi-based human activity recognition. Pattern Recognition, 153, (2024) 110480. https://doi.org/10.1016/j.patcog.2024.110480

[23] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, et al., mHealthDroid: A novel framework for agile development of mobile health applications. In International Workshop on Ambient Assisted Living, Springer, Switzerland, (2014) 91-98. https://doi.org/10.1007/978-3-319-13105-4_14

[24] A. Tehrani, M. Yadollahzadeh-Tabari, A. Zehtab-Salmasi, R. Enayatifar, Wearable sensor-based human activity recognition system employing bi-LSTM algorithm. The Computer Journal, 67(3), (2024) 961-975. https://doi.org/10.1093/comjnl/bxad035

[25] A. Reiss, D. Stricker, (2012) Introducing a new benchmarked dataset for activity monitoring. In 16th International Symposium on Wearable Computers, IEEE, UK. https://doi.org/10.1109/ISWC.2012.13

[26] Z. Wu, Y. Xiong, S. X. Yu, D. Lin, (2018) Unsupervised feature learning via non-parametric instance discrimination. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, USA. https://doi.org/10.1109/CVPR.2018.00393

[27] Y. Tao, K. Takagi, K. Nakata, (2021) Clustering-friendly representation learning via instance discrimination and feature decorrelation. arXiv preprint arXiv.

[28] V. Revathi, B. P. Kavin, A. Thirumalraj, E. Gangadevi, B. Balusamy, S. Gite, Image-based feature separation using RBM tech with ADBN tech for accurate fruit classification. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), IEEE, India. https://doi.org/10.1109/IC2PCT60090.2024.10486564

[29] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional block attention module. In European Conference on Computer Vision (ECCV), Munich, Germany, (2018) 3-19. https://doi.org/10.1007/978-3-030-01234-2_1

[30] Y. Fu, D. Liu, J. Chen, L. He, Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems. Artificial Intelligence Review, 57(5), (2024) 1-102. https://doi.org/10.1007/s10462-024-10729-y

Authors Contribution Statement

Janardhan M: Conceptualization, Methodology, Validation. Neelima A: Software, Implementation. Siri D: Conceptualization, Investigation, Writing - review & editing. Sathish Kumar R: Writing original draft,

Validation. Balakrishna N: Writing original draft. Sreenivasa N: Software, Implementation. Tejesh Reddy Singasani: Methodology, Validation. Ramesh Vatambeti: Methodology, Writing - review & editing.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes

About the License