

## Improved Archimedes Optimization Algorithm with Deep Learning Empowered Fall Detection System

Ala Saleh Alluhaidan<sup>1</sup>, Masoud Alajmi<sup>2</sup>, Fahd N. Al-Wesabi<sup>3,4</sup>, Anwer Mustafa Hilal<sup>5</sup>,  
Manar Ahmed Hamza<sup>5,\*</sup> and Abdelwahed Motwakel<sup>5</sup>

<sup>1</sup>Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O.Box 84428, Riyadh, 11671, Saudi Arabia

<sup>2</sup>Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia

<sup>3</sup>Department of Computer Science, College of Science & Arts at Mahayil, King Khalid University, Muhayel Aseer, 62529, Saudi Arabia

<sup>4</sup>Faculty of Computer and IT, Sana'a University, Sana'a, 61101, Yemen

<sup>5</sup>Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj, 16278, Saudi Arabia

\*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa

Received: 16 November 2021; Accepted: 30 December 2021

**Abstract:** Human fall detection (FD) acts as an important part in creating sensor based alarm system, enabling physical therapists to minimize the effect of fall events and save human lives. Generally, elderly people suffer from several diseases, and fall action is a common situation which can occur at any time. In this view, this paper presents an Improved Archimedes Optimization Algorithm with Deep Learning Empowered Fall Detection (IAOA-DLFD) model to identify the fall/non-fall events. The proposed IAOA-DLFD technique comprises different levels of pre-processing to improve the input image quality. Besides, the IAOA with Capsule Network based feature extractor is derived to produce an optimal set of feature vectors. In addition, the IAOA uses to significantly boost the overall FD performance by optimal choice of CapsNet hyperparameters. Lastly, radial basis function (RBF) network is applied for determining the proper class labels of the test images. To showcase the enhanced performance of the IAOA-DLFD technique, a wide range of experiments are executed and the outcomes stated the enhanced detection outcome of the IAOA-DLFD approach over the recent methods with the accuracy of 0.997.

**Keywords:** Fall detection; intelligent model; deep learning; archimedes optimization algorithm; capsule network

### 1 Introduction

Aging is one of the common problems associated with our life expectancy. The World Health Organization (WHO) says that aging people is 20% population of the world. Other reports say that elderly population (more than 65 years) will be increasing to 1.5 billion by 2050. Generally, older age



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

decreases the overall sensory, physical, and cognitive functions. Thus, elderly people face difficulties in operating day to day activities like dressing up, walking, jogging, and eating [1]. Falling is one of the serious problems in old age people which could decrease their life expectancy. Consequently, it causes injury which could be frequently fatal. Also, Psychological grievances were described as the consequences of falls. People might suffer from depression, fear of falling, anxiety, and activity restriction [2]. The main physiological problem in elderly people is constraining their Activities of Daily Life (ADL), fear of falling. This fear result in activity restriction that leads to weakened muscle and inadequate gait balance which affect the independence and mobility of elderly people [3].

Advancements in signal processing and sensor technologies have become the basis for innovative fall detection (FD) and autonomous activity monitoring system [4]. Sensor information is gathered and processed with signal processing units to gather data regarding the position of the senior. This system comprises variety of sensors i.e., ambient (for example cameras, microphones) or wearable (for example smartwatches, Inertial Measurement Units (IMU) devices, smartphones). In this work, researchers focus more on the usage of IMU sensor which constitutes customized devices or is embedded in smart phones. This sensor belongs to the class of Microelectromechanical sensors (MEMS) which is made on a small scale, and the design integrates a portion which could physically move or vibrate.

In this study, we considered that these problems are approached from multiple perceptions [5]. For example, emphasize the use of the Machine Learning (ML) method employed to video audio processing, and image detection for detecting falls. Especially, we need to highlight that primary goal is the passive monitorization of dependent individuals in opposition to common (and lesser efficient) active system based teleassistance, where the dependent should activate the necklace to request help [6]. This aim can be attained by a low-cost robot and the Wireless Sensor Network (WSN) deployment, caregiving services through a Web-based method or controlled by dependent family members. To attain an effective monitorization, this scheme's aim is to comprise a fall detecting based ML method, collecting the necessary information by means of a tri-axial accelerometer embedding within a bracelet or smartwatch wearied by the dependent people [7].

The major drawback of the traditional telecare system is the psychological rejection of the dependant to other devices could expose his/her dependency [8]. A workaround that is widely employed to offer smartwatches to the dependant, which doesn't have negative connotations and appears to be a regular watch. Those devices have become increasingly complex, involving various healthcare monitoring abilities. Amongst others, the more common are accelerometer, widely employed for monitoring daily tasks like sleep or walking, which is utilized for fall detection [9]. Furthermore, the smartwatch is placed often from the wrist, enabling the advancement of FD algorithms, since they utilize an identical relative posture regarding the body. Despite the comprehensive study about ML methods for fall detecting system, a major challenge faced by all researchers are the small quantity of accurate fall available data sets; appropriate training of ML algorithms needs massive number of information.

This paper presents an Improved Archimedes Optimization Algorithm with Deep Learning Empowered Fall Detection (IAOA-DLFD) model to identify the fall/non-fall events. The proposed IAOA-DLFD technique comprises different levels of pre-processing to improve the input image quality. Besides, the IAOA with Capsule Network (CapsNet) based feature extractor is derived to produce an optimal set of feature vectors. In addition, the IAOA uses to significantly boost the overall FD efficiency by optimal choice of CapsNet hyperparameters. Lastly, radial basis function

(RBF) network was applied for determining the proper class label of the test images. For investigating the improved outcomes of the IAOA-DLFD technique, a series of simulations were performed on benchmark dataset.

## 2 Related Works

Ramirez et al. [10] described the winning technique advanced for Multimodal Fall Detection competition. It can be multiple-sensor data-fusion ML technique which identifies falls and human activities via five wearable inertial sensors: gyroscopes and accelerometers. To effectively adapt the methodology to the three test samples, they implemented unsupervised similarity searches—which detect the 3 common users to the 3 users in the test sample. Mrozek et al. [11] presented a scalable framework which is capable of monitoring fall detection and informing caregivers. The scalability test discloses the requirement to allow larger scaling operation. Furthermore, validate several ML methods for estimating the suitability in the detection system.

Sangeetha et al. [12] presented a recurrent neural network (RNN) based architecture for the detection of daily or fall activities of a person who has neurological disorders and handle the person by concerning the physicians. When an abnormality is identified in the day-to-day activity it will inform family member or caregiver. The RNN based fall detecting method integrated knowledge from the wearable or smartphone and camera deployed on the ceiling and wall. Villaverde et al. [13] designed an open-source fall simulator which could recreate accelerometer fall samples of 2 standard kinds of fall: forward and syncope. This simulated sample is like real fall recorded by the real accelerometer for using them as input for ML application.

Soni et al. [14] presented the fall detecting system with ML method that subsequently employs a fog computing method to transmit data to the caregivers in real-time. Smartphone accelerometers are employed to collect the information and one class technique-based support vector machine (SVM) is employed for building fall detection. Then, consider 5 features from the Smartphone accelerometers to build fall detecting system. Nari et al. [15] goal is to face these problems by presenting a SaveMeNow.AI method, an advanced wearable device for detecting falls. SaveMeNow.AI technique is depending on the deployment of ML method for detecting falls which is embedded in it. This method uses information constantly evaluated by a 6-axis IMU existing within the device.

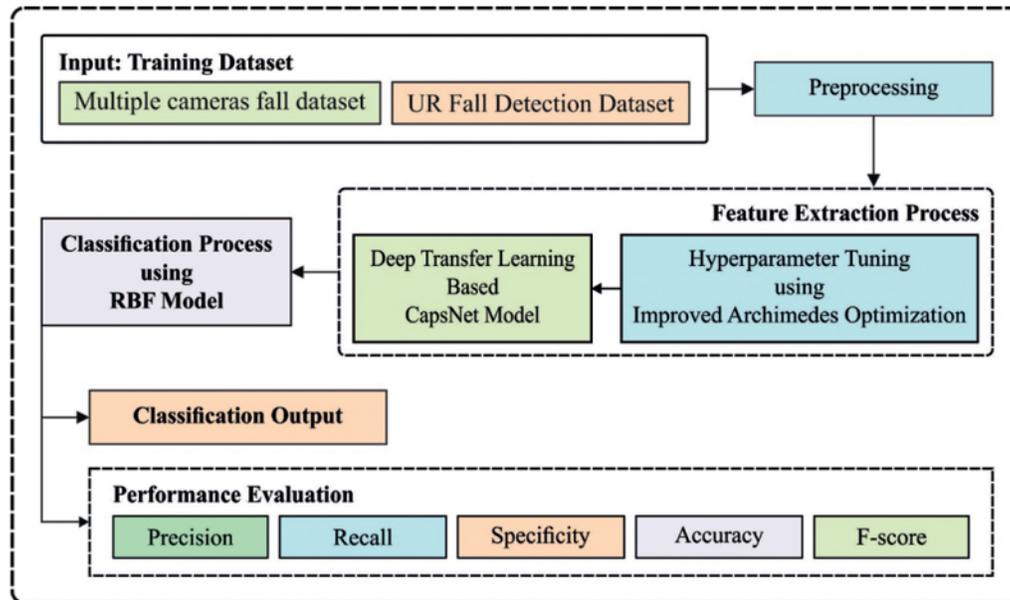
## 3 The Proposed Fall Detection Model

In this study, an effective IAOA-DLFD technique has been developed for fall detection and classification. The proposed IAOA-DLFD technique encompasses pre-processing, Capsule Network based Feature Extraction, IAOA based Hyperparameter tuning and RBF based Classification. Fig. 1 demonstrates the overall working process of IAOA-DLFD technique. The comprehensive working of these procedures is elaborated in the following sections.

### 3.1 Level I: Pre-processing

In the initial step, the frame was pre-processed to increase the features of an image, elimination the noise objects, and enhance particular gets of features. At present, the frame was processed in 3 essential levels such as resize, augmentation, and normalized. For decreasing the computation cost, the resized frame takes place in  $150 \times 150$ . Simultaneously, the frame was augmentation in that the frame was different at each trained epochs. In order to augment the frame, several methods as rotation,

zoom, height shift, width, and horizontal flip. Eventually, normalized approach has been executed for improving generalized of the method.



**Figure 1:** Overall process of proposed IAOA-DLFD technique

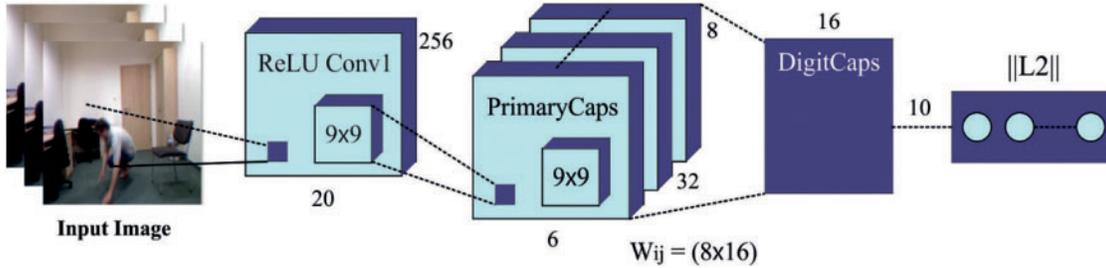
### 3.2 Level II: CapsNet Model

At the second stage, the CapsNet model is utilized to produce a collection of feature vectors. For overcoming the limitations of convolutional neural network (CNN) and generate it nearby the cerebral cortex action framework, Sabour et al. [16] presented a maximal dimension vector named as “capsule” for representing an entity (object/quantity of objects) by set of neurons before single neuron. The actions of neurons in active capsule signify many features of specifying entity which is projected from the image. All the capsules learned an implicit explanation of visual entities which output the possibility of entities and the group of “instantiated parameters containing the precise pose (orientation, place, and size), texture, deformation, albedo, velocity, hue, and so on.

The framework of CapsNet was distinct from other deep learning (DL) techniques. Fig. 2 illustrates the framework of CapsNet. The outcomes of input as well as output of CapsNets are vector, whose standard and way signify the existence probabilities and several attributes of entities correspondingly. A similar level of capsule is used for predicting the instantiation parameter of high level capsule with transformation matrix and, afterward, dynamic routing was implemented for making the forecast consistent. If several forecasts are consistent, the high level of 1 capsule was developed actively.

An easy CapsNet framework was shallow with only 2 convolution layers (Conv1, PrimaryCaps) and 1 fully connected (FC) layer (Entity- Caps). Especially, Conv1 has typical Conv layer that changes images to initial feature and outcomes to PrimaryCaps with Conv filter through a size of  $13 \times 13 \times 256$ . During the case, novel images could not appropriate to an input of 1<sup>st</sup> layer of CapsNet, the main features then Convs are implemented. The 2<sup>nd</sup> Conv layer makes the equivalent vector framework as input of capsule layers. The typical Conv of every outcome was scalar, however, the Conv of

PrimaryCaps has distinct from the classical one. It could be considered as 2D Conv of 8 various weights to an input of  $15 \times 15 \times 256$ . The 3<sup>rd</sup> layer (EntityCaps) has resultant layer that has 9 typical capsules equivalent to 9 various classes.



**Figure 2:** CapsNet architecture

Recall the original Equation, a capsule layer is separated into several computation units called capsules. Consider that the  $i$  capsule output activity vector  $u_i$  from the PrimaryCaps  $i$ , it can be given to  $j$  capsule for generating activity level  $v_j$  of EntityCaps. Updating and Propagation are carried out by the vectors among EntityCaps and PrimaryCaps. Matrix processing is employed to the scalar input in all the layers of CNN, i.e., basically a linear integration of output. The capsule processing input has been separated into 2 phases: routing and linear integration. Here, the linear integration represents the concept of process scalar input by NN that implies processing the relationships among 2 objects from the scene via the visual conversion matrix when maintaining their similar relationship [17]. Especially, the linear integration can be expressed as follows

$$\hat{u}_{ji} = u_i W_{ij} \quad (1)$$

Let  $\hat{u}$  be a predictive vector generated by converting the output  $u_i$  of capsule from the below layer by the weight  $W_{ij}$ . Next, during the routing phase, the  $s_j$  input vector of  $j$  capsule is determined by Eq. (2)

$$s_j = \sum_i c_{ij} \hat{u}_{ji} \quad (2)$$

In the equation,  $c_{ij}$  represent the coupling coefficients defined by iteration dynamic routing. Actually, the routing has been weighted summation of  $\hat{u}$  with the coupling coefficients.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}. \quad (3)$$

Apparently, the capsule activation function essentially redistributes and suppresses vector length. Its output is employed by the likelihood of entity denoted as the capsule in the present class.

The overall loss function of original CapsNet has been weighted sum of reconstruction loss and marginal loss. The MSE was employed in the original recreation loss function that significantly degrade the method while processing noisy information.

### 3.3 Level III: IAOA Based Hyperparameter Tuning

At this stage, the hyperparameters of the CapsNet model are optimally tuned by the use of IAOA. AOA technique was simulated in the standard of Archimedes that assumed as law of physics. The Archimedes rule was involved in the object that was partial or complete immersed from the fluids.

Because it can upward force (named as buoyancy) created in the liquid on the body, this force was equivalent to fluid weight moved in the body. In AOA, the immersed object was assumed that the population individual (candidate solution) [18]. This method starts with initializing a population with objects, also the place of all objects is initialization from arbitrary method from the issue search space. Next, the equivalent fitness function (FF) was computed. In the iteration procedure, AOA upgrades the object density as well as volume but its acceleration was upgraded on the fundamental of its collision with neighbor objects. The initialized procedure of every object was implemented utilizing the subsequent equation:

$$O_i = l_i + rand \times (u_i - l_i), \quad i = 1, 2, \dots, N \quad (4)$$

where  $l_i$  and  $u_i$  refers the lower as well as upper limits of  $i^{th}$  objects and  $N$  signify the count of objects. The volumes as well as densities of all objects are initialized as:

$$den_i = rand, \quad vol_i = rand \quad (5)$$

where  $rand$  stands for the vector of  $D$  dimensional with values from the range of zero and one. The acceleration of all objects is computed as [19]:

$$acc_i = lb_i + rand \times (ub_i - lb_i) \quad (6)$$

The primary FF was computed and object with optimum fitness has allocated as  $x^{best}$ ,  $den^{best}$ ,  $vol^{best}$ , and  $acc^{best}$ .

The upgrading procedure of  $i^{th}$  object densities as well as volumes are implemented dependent upon the subsequent equation:

$$den_i^{t+1} = den_i^t + rand \times (den^{best} - den_i^t) \quad (7)$$

$$vol_i^{t+1} = vol_i^t + rand \times (vol^{best} - vol_i^t) \quad (8)$$

where  $t$  implies the present iterative and  $rand$  signifies the arbitrary number. Initially, there exist collision among the object then that object attempts for reaching the equilibrium state. This performance was projected in AOA using transfer function that uses from alteration in exploration to exploitation stages.

The equation of transfer function was expressed as:

$$TF = \exp \left( \frac{t - t_{max}}{t_{max}} \right) \quad (9)$$

where  $t_{max}$  refers the maximal count of iterations. At this point, the value of  $TF$  has enhanced slowly with iterative still it attains to unity. The density reducing issue is another one which uses AOA to transfer in global to local searches, it could be expressed as:

$$d^{t+1} = \exp \left( \frac{t - t_{max}}{t_{max}} \right) - \left( \frac{t}{t_{max}} \right) \quad (10)$$

The value of  $d^{t+1}$  reduces with time, also appropriate allocating of this variable uses from obtaining exploration or exploitation balance. The exploration stage was demonstrated as collision amongst the objects, this stage was regarded if the transfer operator has 0.5. The acceleration of  $i^{th}$  object at iterative  $t + 1$  has upgraded by chosen random material ( $mr$ ) as:

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \quad (11)$$

where  $n_{mr}$ ,  $vol_{mr}$ , and  $acc_{mr}$  represents the density, volume, and acceleration of  $mr$ . The acceleration of  $i^{th}$  objects from exploitation stage was calculated as:

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \quad (12)$$

where  $n_{best}$ ,  $vol_{best}$ , and  $acc_{best}$  stands for the optimum object density, volume, and acceleration correspondingly. It can be essential for normalizing the acceleration of all particles, this defines the step percentage by which all the particles are modified. The normalization acceleration was expressed as:

$$acc_{i-norm}^{t+1} = u \times \left( \frac{acc_i^{t+1} - \min(acc)}{\max(acc) - \min(acc)} \right) + l \quad (13)$$

where  $l$  and  $u$  imply the normalized ranges, it can be allocated as 0.1 and 0.9 correspondingly. If the object is distant in the global optimum, the value of accelerations is maximum, during this study exploration stage has shown else exploitation step was projected.

The place of  $i^{th}$  particle has upgraded from exploration stage utilizing the subsequent equation:

$$x_i^{t+1} = x_i^t + C_1 \times rand \times acc_{i-norm}^{t+1} \times d \times (x_{rand} - x_i^t) \quad (14)$$

Conversely, the upgrade method of particle places under the exploitation stage was projected as:

$$x_i^{t+1} = x_i^t + F \times C_2 \times rand \times acc_{i-norm}^{t+1} \times d \times (T \times x_{best} - x_i^t) \quad (15)$$

where  $C_1$  and  $C_2$  represents the constants determined as users,  $T$  implies the parameter which dependent upon transfer operator ( $T = C_3 \times TF$ ),  $C_3$  signifies the constant value,  $x_{best}$  refers the place of an optimum particle, and  $F$  signifies the flag utilized for changing the particle motion directions. The value of  $F$  has defined utilizing the subsequent equation:

$$F = \begin{cases} +1P \leq 0.5 \\ -1P > 0.5 \end{cases} \quad (16)$$

where the value of  $P$  has been allocated arbitrarily by user. Lastly, the FF was calculated at the upgraded particle places and afterward, the optimum solutions are recorded.

Some meta-heuristic techniques obtainable still date undergoes premature convergence or are ineffective for attaining global optimum in few cases. Therefore, improving the exploration abilities of present AOA is castoff the mutation operators [20]. It supports balanced motion from present and opposite direction with minimum and maximum mutations according to the optimal position. The normal distributed mutations process has beneficial. The normally distributed mutation produces a single mutant as represented in Eq. (17).

$$\begin{aligned} A_i^y &= P_i + N(0, \sigma^2) \\ &= P_i + \sigma * N(0, 1) \end{aligned} \quad (17)$$

where  $P_i$  refers the un-mutated Archimedes population and  $\sigma$  represents the mutated factor on chosen dimensional. Therefore place of every  $i^{th}$  Archimedes from  $y^{th}$  dimension are upgraded as:

$$P_i^y = P_i + [SRD * P_i * N(0, 1)] \quad (18)$$

where  $SRD \in [0.2, 0.4, 0.6, 0.8, 1]$ .

### 3.4 Level IV: RBF Network Based Classification

Finally, the RBF network is utilized as a classification model to allot proper class labels to it. RBF network comprises output, input, and hidden layers. The network is feed-forward network. The functionality of the input layer is similar to other networks that are to take inputs and provide outputs, the main distinction for other networks lies in the functioning of hidden layer. During this method, the hidden layer comprises a certain activation function called RBF. The RBF is named to be certain classes of linear function which have a unique feature, where the response increases/reduces monotonically with distance from a centre point. The hidden layer was accountable for performing nonlinear conversion of output and input layer carries out linear regression to predict the output. RBF is distinct from another network which has many hidden layers active at a time. Multiquadric RBF has the properties of monotonically increasing with the distance from the centre and a Gaussian RBF have the properties of monotonically decreasing with distance from the centre.

## 4 Performance Validation

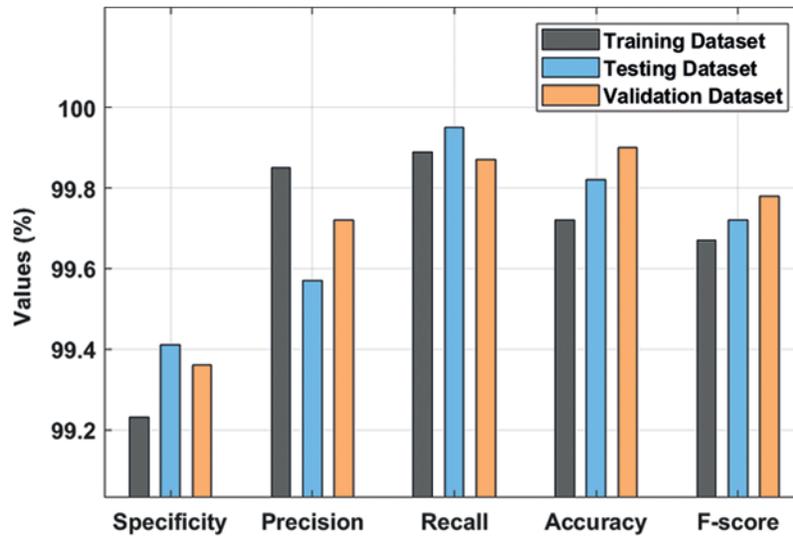
The experimental validation of the IAOA-DLFD technique can be validated using URFD (URFD) [21] and Multiple Cameras Fall (MCF) [22] datasets. The MCF dataset comprises 192 videos including 96 fall and 96 indoor activities. In addition, the URFD dataset includes frontal and overhead video sequences with 314 and 302 frontal and overhead sequences respectively.

A brief examination of the IAOA-DLFD technique under training dataset (TD), testing dataset (TSD), and validation dataset (VD) on MCF dataset is given in Tab. 1 and Fig. 3. The outcomes show that the IAOA-DLFD technique has offered enhanced fall detection outcomes on all data. For instance, with TD, the IAOA-DLFD technique has resulted to  $SPEC_y$ ,  $PREC_n$ ,  $REC_l$ ,  $ACC_y$ , and  $F_{SCORE}$  of 99.23%, 99.85%, 99.89%, 99.72%, and 99.67% respectively. Besides, the IAOA-DLFD technique has attained maximum average  $SPEC_y$ ,  $PREC_n$ ,  $REC_l$ ,  $ACC_y$ , and  $F_{SCORE}$  of 99.33%, 99.71%, 99.90%, 99.81%, and 99.72% respectively.

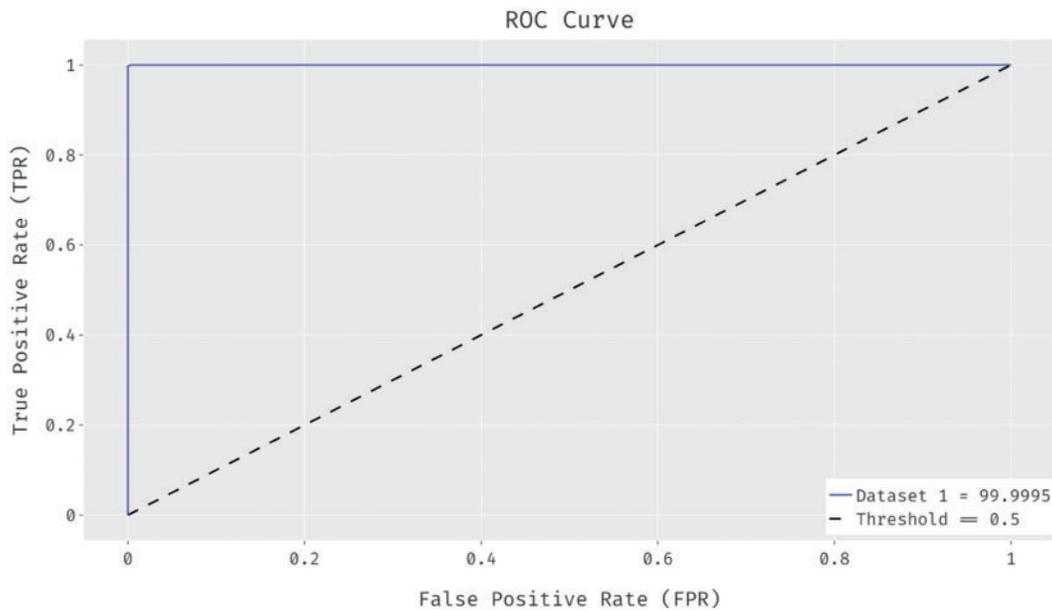
**Table 1:** Analysis of IAOA-DLFD model on MCF dataset

Dataset	Specificity	Precision	Recall	Accuracy	F-score
Training dataset	99.23	99.85	99.89	99.72	99.67
Testing dataset	99.41	99.57	99.95	99.82	99.72
Validation dataset	99.36	99.72	99.87	99.90	99.78
Average	99.33	99.71	99.90	99.81	99.72

Fig. 4 demonstrates the ROC analysis of the IAOA-DLFD technique on the test MCF dataset. The figure revealed that the IAOA-DLFD technique has gained enhanced outcome with the maximum ROC of 99.9995.



**Figure 3:** Result analysis of IAOA-DLFD technique on MCF dataset



**Figure 4:** ROC graph of AOA-DLFD technique on MCF dataset

A detailed examination of the IAOA-DLFD approach under TD, TSD, and VD on URFD dataset are offered in [Tab. 2](#) and [Fig. 5](#). The outcomes demonstrated that the IAOA-DLFD method has obtainable improved fall detection outcomes on all data. For sample, with TD, the IAOA-DLFD method has resulted to  $SPEC_y$ ,  $PREC_n$ ,  $REC_l$ ,  $ACC_y$ , and  $F_{SCORE}$  of 99.58%, 99.95%, 99.63%, 99.54%, and 99.41% correspondingly. In addition, the IAOA-DLFD system has reached maximal average  $SPEC_y$ ,  $PREC_n$ ,  $REC_l$ ,  $ACC_y$ , and  $F_{SCORE}$  of 99.59%, 99.79%, 99.61%, 99.62%, and 99.44% correspondingly.

**Table 2:** Analysis of IAOA-DLFD model on URFD dataset

Dataset	Specificity	Precision	Recall	Accuracy	F-score
Training dataset	99.58	99.95	99.63	99.54	99.41
Testing dataset	99.64	99.63	99.42	99.61	99.24
Validation dataset	99.54	99.78	99.77	99.72	99.67
Average	99.59	99.79	99.61	99.62	99.44

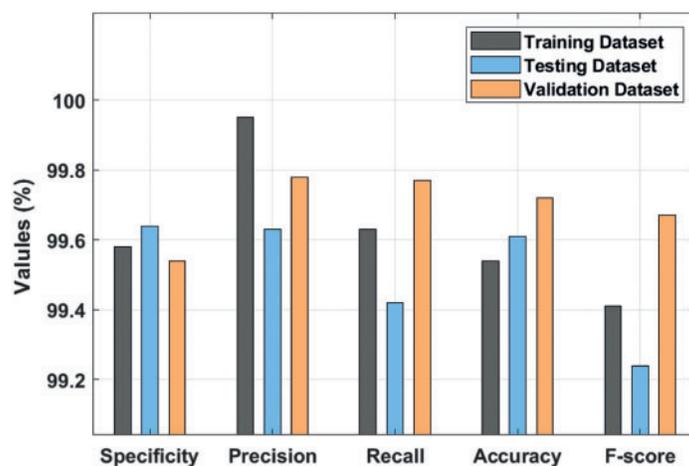
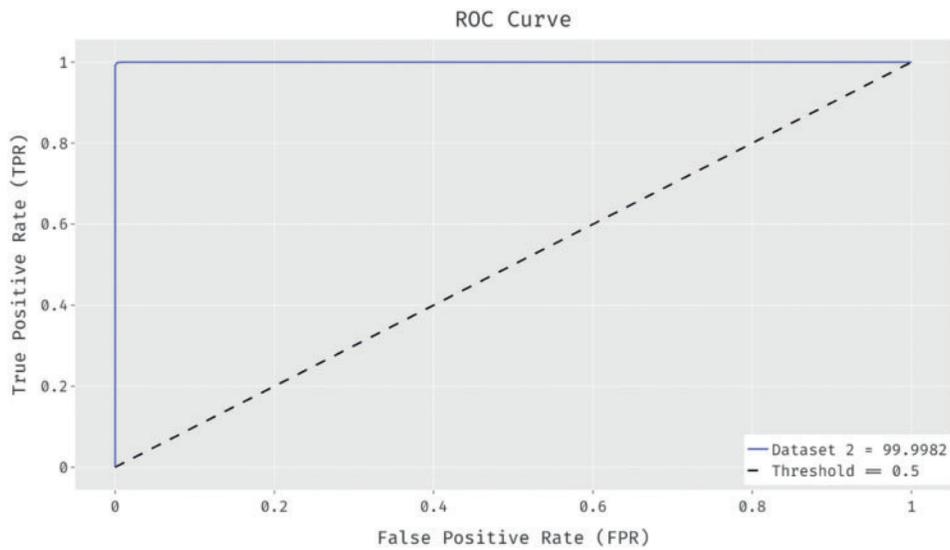
**Figure 5:** Result analysis of IAOA-DLFD technique on URFD dataset

Fig. 6 depicts the ROC analysis of the IAOA-DLFD manner on the test URFD dataset. The figure exposed that the IAOA-DLFD technique has attained improved outcomes with the maximal receiver operating characteristic (ROC) of 99.9982.

A comparative  $ACCU_y$  analysis of the IAOA-DLFD technique on MCF dataset is offered in Tab. 3. The outcomes revealed that the 2Dimen\_CNN, 1Dimen.\_CNN, RNet\_50 and RNet\_101 models have obtained minimal  $ACCU_y$  values of 0.955, 0.943, 0.961, and 0.965 respectively. In line with, the DL\_VGG\_16, DL\_VGG\_19, and Exception techniques have obtained moderately closer  $ACCU_y$  values of 0.980, 0.980, and 0.978 respectively. Through the IoT enabled elderly fall detection model using optimal deep convolutional neural network (IMEFD-ODCNN) technique has accomplished competitive  $ACCU_y$  of 0.998, the presented IAOA-DLFD technique has gained maximum  $ACCU_y$  of 0.999.

A comparative  $ACCU_y$  analysis of the IAOA-DLFD manner on URFD dataset is obtainable in Tab. 4. The outcomes stated that the 1D\_Conv\_NN, 2D\_Conv\_NN, ResNet\_50 and ResNet\_101 approaches have reached lesser  $ACCU_y$  values of 0.927, 0.950, 0.954, and 0.962 correspondingly. Followed by, the VGG\_16, DL\_VGG\_19, and Depthwise techniques have reached moderately closer  $ACCU_y$  values of 0.976, 0.980, and 0.980 respectively. But, the IMEFD\_ODCNN methodology has accomplished competitive  $ACCU_y$  of 0.996, the projected IAOA-DLFD approach has reached maximal  $ACCU_y$  of 0.997.



**Figure 6:** ROC graph of AOA-DLFD technique on URFD dataset

**Table 3:** Comparative accuracy study of IA OA-DLFD technique on MCF dataset

Methods	Accuracy
DL_VGG_16	0.980
DL_VGG_19	0.980
Exception model	0.978
1Dimen._CNN	0.943
2Dimen._NN	0.955
RNet_50	0.961
RNet_101	0.965
IMEFD_ODCNN	0.998
IAOA-DLFD	0.999

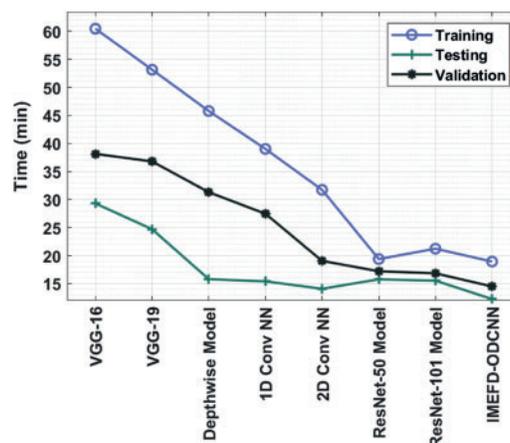
**Table 4:** Comparative accuracy study of IA OA-DLFD technique on URFD dataset

Methods	Accuracy
VGG_16	0.976
VGG_19	0.980
Depthwise	0.980
1D_Conv_NN	0.927
2D_Conv_NN	0.950
ResNet_50	0.954
ResNet_101	0.962
IMEFD_ODCNN	0.996
IAOA-DLFD	0.997

The time complexity analysis of the IAOA-DLFD technique on the MCF dataset in terms of training time (TT), testing time (TST), and validation time (VT) is shown in Tab. 5 and Fig. 7. The results show that the VGG\_16 and VG\_19 manners have shown ineffective outcomes with the maximum time complexity. At the same time, the Depthwise, 1D\_Conv\_NN, and 2D\_Conv\_NN approaches have obtained moderately reduced time complexity. Moreover, the ResNet\_50 and ResNet\_101 systems have attained reasonably low time complexity. However, the presented IAOA-DLFD technique has resulted in improved outcomes with the least TT, TST, and VT of 18.949, 12.257, and 14.498 min respectively.

**Table 5:** Time complexity analysis of IAOA-DLFD technique on MCF dataset

Models	TT (min)	TST (min)	VT (min)
VGG_16	60.458	29.306	38.142
VGG_19	53.151	24.708	36.796
Depthwise model	45.786	15.806	31.310
1D_Conv_NN	39.023	15.410	27.469
2D_Conv_NN	31.724	14.094	19.061
ResNet_50 model	19.391	15.770	17.213
ResNet_101 model	21.236	15.540	16.844
IAOA-DLFD	18.949	12.257	14.498



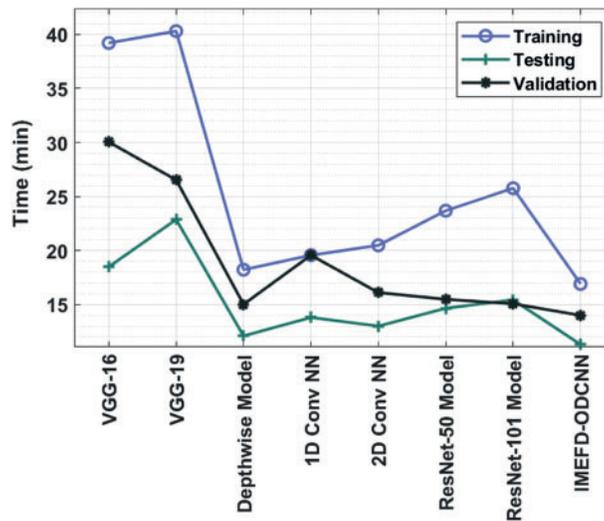
**Figure 7:** Time complexity analysis of IAOA-DLFD approach in MCF dataset

The time complexity analysis of the IAOA-DLFD system on the URFD dataset with respect to TT, TST, and VT is illustrated in Tab. 6 and Fig. 8. The outcomes exhibited that the VGG\_16 and VG\_19 manners have outperformed ineffective outcomes with the higher time complexity. Likewise, the Depthwise, 1D\_Conv\_NN, and 2D\_Conv\_NN models have obtained moderately minimum time complexity. In addition, the ResNet\_50 and ResNet\_101 methods have attained reasonably low time complexity. Eventually, the presented IAOA-DLFD approach has resulted in enhanced outcomes with the worse TT, TST, and VT of 16.900, 11.290, and 13.995 min correspondingly. From the above

mentioned results analysis, the IAOA-DLFD technique can be utilized as a novel tool to FD and classification.

**Table 6:** Time complexity analysis of IAOA-DLFD technique on URFD dataset

Methods	Training time (s)	Testing time (s)	Validation time (s)
VGG_16	39.210	18.480	30.042
VGG_19	40.310	22.870	26.534
Depthwise model	18.220	12.090	15.013
1D_Conv_NN	19.560	13.800	19.558
2D_Conv_NN	20.480	13.000	16.099
ResNet_50 model	23.680	14.650	15.479
ResNet_101 model	25.760	15.430	15.064
IAOA-DLFD	16.900	11.290	13.995



**Figure 8:** Time complexity analysis of IAOA-DLFD approach in URFD dataset

## 5 Conclusion

In this study, an effective IAOA-DLFD technique has been developed to FD and classification. The proposed IAOA-DLFD technique encompasses pre-processing, capsule network based feature extraction, IAOA based hyperparameter tuning, and RBF based classification. In addition, the IAOA uses for significantly boost the overall FD performance by optimal choice of CapsNet hyperparameters. In order to showcase the enhanced performance of the IAOA-DLFD technique, a wide range of experiments are implemented and the outcomes stated the enhanced detection outcome of the IAOA-DLFD technique over the recent methods under several aspects. Therefore, the IAOA-DLFD technique can be utilized as an effective tool for fall detection and classification. As a part of future scope, multimodal fusion models can be developed to boost fall detection outcomes.

**Funding Statement:** This work was supported by Taif University Researchers Supporting Program (Project Number: TURSP-2020/195), Taif University, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under Grant Number (RGP 2/209/42). Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2022R234), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] A. Bhattacharya and R. Vaughan, "Deep learning radar design for breathing and fall detection," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 5072–5085, 2020.
- [2] N. Thakur and C. Y. Han, "A study of fall detection in assisted living: Identifying and improving the optimal machine learning method," *JSAN*, vol. 10, no. 3, pp. 39, 2021.
- [3] I. Sreenidhi, "Real-time human fall detection and emotion recognition using embedded device and deep learning," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 3, pp. 780–786, 2020.
- [4] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros and B. J. T. Fernandes, "A multimodal approach using deep learning for fall detection," *Expert Systems with Applications*, vol. 168, pp. 114226, 2021.
- [5] A. Sultana, K. Deb, P. K. Dhar and T. Koshiba, "Classification of indoor human fall events using deep learning," *Entropy*, vol. 23, no. 3, pp. 328, 2021.
- [6] F. S. Khan, M. N. H. Mohd, D. M. Soomro, S. Bagchi and M. D. Khan, "3D hand gestures segmentation and optimized classification using deep learning," *IEEE Access*, vol. 9, pp. 131614–131624, 2021.
- [7] E. Torti, A. Fontanella, M. Musci, N. Blago, D. Pau *et al.*, "Embedded real-time fall detection with deep learning on wearable devices," in *2018 21st Euromicro Conf. on Digital System Design (DSD)*, Prague, Czech Republic, pp. 405–412, 2018.
- [8] T. Mauldin, M. Canby, V. Metsis, A. Ngu and C. Rivera, "SmartFall: A smartwatch-based fall detection system using deep learning," *Sensors*, vol. 18, no. 10, pp. 3363, 2018.
- [9] T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. Diaz, I. V. Pustokhina *et al.*, "Internet of things and deep learning enabled elderly fall detection model for smart homecare," *IEEE Access*, vol. 9, pp. 113879–113888, 2021.
- [10] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris *et al.*, "Fall detection and activity recognition using human skeleton features," *IEEE Access*, vol. 9, pp. 33532–33542, 2021.
- [11] D. Mrozek, A. Koczur and B. M. Mrozek, "Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge," *Information Sciences*, vol. 537, pp. 132–147, 2020.
- [12] D. P. Sangeetha, "Fall detection for elderly people using video-based analysis," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. SP7, pp. 232–239, 2020.
- [13] A. C. Villaverde, M. Cobos, P. Muñoz and D. F. Barrero, "A simulator to support machine learning-based wearable fall detection systems," *Electronics*, vol. 9, no. 11, pp. 1831, 2020.
- [14] M. Soni, S. Chauhan, B. Bajpai and T. Puri, "An approach to enhance fall detection using machine learning classifier," in *2020 12th Int. Conf. on Computational Intelligence and Communication Networks (CICN)*, Bhimtal, India, pp. 229–233, 2020.
- [15] M. I. Nari, S. S. Suprpto, I. H. Kusumah and W. Adiprawita, "A simple design of wearable device for fall detection with accelerometer and gyroscope," in *2016 Int. Symp. on Electronics and Smart Devices (ISESD)*, Bandung, Indonesia, pp. 88–91, 2016.
- [16] S. Sabour, N. Frosst and G. Hinton, "Dynamic routing between capsules," arXiv preprint arXiv:1710.09829, 2017.
- [17] F. Deng, S. Pu, X. Chen, Y. Shi, T. Yuan *et al.*, "Hyperspectral image classification with capsule network using limited training samples," *Sensors*, vol. 18, no. 9, pp. 3153, 2018.

- [18] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk and W. Al-Atabany, “Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems,” *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.
- [19] A. Fathy, A. Alharbi, S. Alshammari and H. Hasanien, “Archimedes optimization algorithm based maximum power point tracker for wind energy generation system,” *Ain Shams Engineering Journal*, vol. 14, pp. 1–14, 2021. <https://doi.org/10.1016/j.asej.2021.06.032>.
- [20] N. Panda and S. K. Majhi, “Oppositional salp swarm algorithm with mutation operator for global optimization and application in training higher order neural networks,” *Multimedia Tools and Applications*, vol. 80, 2021. <https://doi.org/10.1007/s11042-020-10304-x>.
- [21] Dataset: <http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html>. 2018.
- [22] Auvinet, C. Rougier, J. Meunier, A. S. Arnaud, J. Rousseau, “MCF dataset,” Technical Report 1350, DIRO-Université de Montréal, 2010.