**Performance Enhancement of Vision Based Fall Detection Using Ensemble of Machine Learning Model**

**1\*Shikha Rastogi, 2Dr. Jaspreet Singh**

1\*Research Scholar, Department Of Computer Science Gd Goenka University, Sohna, Haryana 122103, India.

2Associate Professor, Department Of Computer Science Gd Goenka University, Sohna, Haryana 122103, India.

Corresponding authors mail id: phd.shikharastogi01@gmail.com

**Abstract**

An automatic fall detection system (FDS) can predict a fall and non-fall to help the elderly by providing timely medical assistance that prevent serious injuries and death. However, most of the existing automatic FDS systems are single networks, which inaccurately detect the falls due to overfitting problems. Nowadays, a few ensemble models are also used to detect falls, but these approaches lack optimal classifiers. So, ensemble based FDS is proposed to improve the fall detection accuracy by choosing the optimal classifier with a greedy algorithm based majority voting approach. In this ensemble model, four machine learning models, namely Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Decision Tree, and Deep LSTM are used. The greedy algorithm based majority voting employs a search-based approach, namely forward search, backward search, and recovery search, with the objective of selecting the optimal classifiers. Forward search adds the classifiers with low majority voting error (MVE) and less processing time. Also, the backward search removes the classifier with low-performance and loss during backward search is avoided using recovery search. Finally, the performance of the proposed ensemble model is compared with the individual ML model and conventional approaches in terms of accuracy, sensitivity, specificity, and precision. The results show that the proposed ensemble model achieves high accuracy (99.98%), sensitivity (99.80%) and specificity (99.90%) compared with the individual ML model. Also, it achieves high accuracy (99.98%), sensitivity (99.80%), specificity (99.90%), and precision (96.23%) compared with conventional approaches.

**Keywords:** Ensemble, Vision based fall detection, Greedy Algorithm, Majority Voting, Selection criteria

**1. Introduction**

The World Health Organisation (WHO) study states that more than 28% of the elderly aged over 65 suffer from falls every year. The elderly population will be 1.2 billion by 2025 around the world, and at the same time, 2.8 million older people have been reported to be suffering from fall injuries [1-4]. Falls can cause serious injuries when impact with the floor or any hard surface is high [5]. A fall is a serious threat to elderly people, causing impacts in terms of physical and financial problems. It also causes serious injuries, hospitalization, and even death for the elderly [6]. The aftermath of fall cases causes health problems such as weak muscles and a decrease in quality of life in the elderly [7]. It is reported that the majority of the elderly who were seen lying on the floor for more than one hour were reported as dead within six months [8]. With the increase in the population of the elderly and the development in the healthcare domain, there is a demand for real-time accurate models to detect the fall [9]. Automatic fall detection collects data, processes it, distinguishes between fall and non-fall events, and alerts rescuers to the victim's location. It can detect falls without human intervention and provides a way to have a safe and independent life for the elderly in their home [1], [4]. Fall data and activity of daily living (ADL) can be collected from ambient sensors, cameras, Wifi, and radar [4]. The vision-based approaches analyse the frames collected from cameras. The ambient sensors record the vibrations from sensors installed on the floor or rooms, and the wearable-based approaches collect the sensor signals from the person’s body through sensors like accelerometers, gyroscopes, and orientation sensors connected to the person’s body [10-12]. The vision-based cameras have proved to be more accurate than sensor-based approaches in the detection of falls. The vision-based approaches use one or multiple cameras to detect the falls. The cameras can be installed in several environments to have surveillance on the elderly. They provide rich visual information about the person and their activities. While comparing the radar and Wifi-based FDS, the vision-based approaches are considered to be cheap. Machine learning approaches have been used in FDS due to their capability to classify fall from non-fall [1], [13], [14]. Supervised machine learning models like support vector machine, hidden markov model, random forest, and K nearest neighbour are widely used for fall detection. Deep learning approaches like Convolutional Neural Network (CNN) help to improve the problems faced by computer vision models. The DL approaches are widely used in image processing approaches due to their automatic feature extraction capability and improved classification accuracy in identifying fall from ADL [15]. Several deep learning approaches like CNN, long short-term memory (LSTM), and auto encoders are employed in the detection of falls with a higher detection rate. Since vision-based approaches face image noise, occlusion, and segmentation incorrectness, some deep learning models have been combined, such as a combined LSTM and CNN model used in vision-based fall detection [4].

It is commonly known that a single network has attained a high degree of classification accuracy. However, the accuracy of these approaches is affected by the overfitting problem. In order to eliminate overfitting and boost detection accuracy, researchers have recently expanded the number of datasets. The networks that can perform well with massive datasets typically have sophisticated structural designs. Complex structures may lengthen training and testing periods, which will impact the algorithm’s real-time performance. The ensemble learning approach, which integrates several networks for training, has been proposed as a solution to this issue. In the ensemble learning paradigm for neural networks, a limited number of networks are trained sequentially on one or more datasets to address the same issue.

The ensemble classifier has the ability to perform better than the individual classifiers. In ensemble learning, individual classification approaches are combined to classify new instances [16]. The diversity of the classifiers and the avoidance of over-fitting due to the control of individual classifiers are some of the advantages of ensemble models. In fall detection, ensemble approaches such as SVM and random forest have demonstrated classification accuracy up to 6% higher than individual classifiers such as SVM [14].

In ensemble learning, it is considered that all classifiers who are regarded as base learners have the same amount of separation power. This consideration is incorrect because different classifiers may exhibit a range of potential behaviours and capabilities depending on the issues and the nature of the information. The main drawback of the ensemble learning approach is that it does not differentiate between various base learners with various resolutions; instead, they are all automatically considered as being at the same level, despite the fact that the resolution of various base learners differs in various environments [17]. These existing ensemble approaches also suffer from low processing time and lack optimal classifiers for accurate fall detection [18].

To avoid these kinds of issues, ensemble based fall detection is proposed based on greedy based majority voting technique. The proposed ensemble based fall detection is based on greedy based majority voting technique. The ensemble is made up of multiple classifier’s and the optimal classifiers are chosen with the greedy based majority voting. The proposed voting based ensemble approach focus to select the optimal classifiers through the selection criteria and fuse them to obtain the classification result. It utilises N-best classifier performance, majority voting improvement (MVI), goodness of a classifier, and majority voting error as the selection criteria. The approach differs from weighted majority where each of the classifier has to be tuned with separate weights. The weight based approach cannot be applied to domain which possess dynamic behaviour [19]. Likewise, the average of probability is carried where voting with the final class is chosen by averaged and maximum average selection. ***The major contributions of the approach are:***

* A vision-based FDS is designed with an ensemble of classifiers, namely SVM, KNN, Decision Tree, and Deep LSTM, to improve the fall detection accuracy.
* A greedy-based majority voting approach is designed with the selection criteria of the classifiers. In greedy based majority voting, forward, backward, and recovery search are modelled with the objective of selecting the best of the classifiers.
* The forward search adds the optimal classifiers based on the selection criteria goodness of the classifier and with the reduced majority voting error (MVE). While the backward search focuses on removing the classifiers based on the criteria of majority voting improvement (MVI). The recovery search is handled with the focus of preserving the loss of any optimal classifiers due to the backward search.
* The proposed ensemble based FDS performance is compared with the individual classifiers and conventional approaches in terms of Accuracy, Precision, Sensitivity and Specificity.

The structure of the paper is as follows. In Section 2, fall detection-based existing techniques are reviewed. Then the proposed ensemble model is introduced in Section 3. Section 4 describes the experiments and results, and Section 5 concludes with recommendations for future research.

**2. Related Works**

One of the top issues in the public healthcare sector is fall detection. Several studies have been conducted on automatic FDS. This section includes a review of these FDS technologies, which is more conceptually similar to the work presented in this paper.

Cai et al. [20] proposed a fall detection strategy with a multi-task hourglass convolutional auto encoder (MTHCAE). Though deep neural networks are widely used in FDS due to the ability to learn the features automatically, the problem of information loss cannot be ruled out in deep neural networks (DNN). The approach made improvements with the inclusion of hour glass residual units (HRU) in the encoder part of MTHCAE. The HRU also helped to extract the multiscale features by capturing more information with fewer convolutional layers. These extracted multiscale features further assist in assuring the high accuracy of fall detection. This method avoids information loss and also performs well with a shallow layer network. This approach prevents information loss and also works well with a shallow layer network.

Espinosa et al. [15] presented a video-based approach for fall detection with CNN. The approach used multiple cameras to collect the fall events. The feature extraction is handled with the optical flow method (OFM), which provides the information connected with the movements in the image. The image pixels horizontal and vertical relative movements are obtained from the image pixels with this approach. The experimental results gained an upper hand over conventional approaches as well as reduced the computation time.

Nogas *et al.* [21] presented a deep fall approach which models the fall detection as anomaly detection problem. The Deep Fall approach uses a deep spatio-temporal convolutional auto encoders [DSTCAE] for this purpose. The unseen falls are detected with an anomaly scoring approach. The deep fall approach is tested with datasets collected from different cameras. It is not realistically possible to find reconstruction error values for future reconstruction errors. So un-normalized reconstruction error is used to assign an anomaly score for a given frame in this approach.

Li et al. [9] proposed a fall detection method that can reduce interference caused by complex background. The saliency map is generated through M-level segmentation. The saliency map is generated with two-stream CNN, which extracts the global and local features. A simple deep network is employed to identify the fall and non-fall with the help of discriminant features. The new approach outperformed existing approaches in terms of accuracy. In this approach, saliency maps limit the influence of complex backdrops, and the preserved features are mostly focused on human activities. When compared to depth images and RGB human blocks, the convergence and accuracy of FDS based on fused saliency maps are enhanced.

Yacchirema et al. [22] presented a fall identification approach with ensemble machine learning (FIEL) in an IoT environment. The elderly moments are collected with a 3D axis accelerometer in an IoT system. Four machine learning algorithms, namely decision trees, deep nets, logistic regression, and ensemble, are used to evaluate the approach. The ensemble-based approaches are used to predict the fall from the data collected by the accelerometer. The best of the models is selected from the performances and the activities are classified into fall and ADL.

Xiong et al. [8] proposed a skeleton-based 3D consecutive-low-pooling neural network (S3D-CNN) to detect the falls. The fall detection model is based on the Long Short Term Memory (LSTM) and gated recurrent unit (GRU) model. The efficiency of a machine learning model is increased with position normalisation and joint point loss. The approach achieved an accuracy of 98.2%, which is better than the baseline model, which obtained 88.9%. By removing background noise and human skeleton sequences beyond the fall interval, this technique optimises sequences from depth videos for fall detection by extracting their movement characteristics.

Jahanjoo et al. [23] presented a fall detection model based on a deep belief network (DBN) approach. The smartphones are embedded with tri-axial acceleration sensors to measure the person’s movements. The DBN is used for training and testing the system. The sensor signals are divided into a series of windows and the noise is removed with a median filter. The features are extracted from the sensor signals with statistical approaches. The DBN classifies the sensor signals. The DBN trains and tests the system with the datasets and classifies the sensor signals into fall and activities of daily living. Different sorts of falls are accurately identified by this technique, using various different features.

Mehta et al. [2] presented a fall detection approach by solving privacy issues and class imbalance problems due to a lower incidence of falls. With the adversarial network, the fall detection is treated as an anomaly detection. Two-channel 3D convolutional auto encoders (2C3DCAE) are used to reconstruct the thermal image. The region-based difference constraint tracks the region of interest (ROI). The reconstruction error is computed by the joint discriminator. Performance is improved by the addition of person-ROI and difference loss functions. The rise in the AUC of the recall-precision curve represents the main advancement over earlier techniques.

Jacome et al. [18] presented a deep learning-based FDS approach. The fall data is collected from IoT-based accelerometer sensors and deep learning models that distinguish fall from non-fall. The fog nodes store the collected sensor data. The deep learning models are deployed on the edge nodes. The proposed approach manages the DL models in memory-scarce fog nodes by using resources through visualisation technologies. Predictive analysis is placed on the fog, near IoT sensors, to reduce the time it takes to identify abnormalities. It also increases the accuracy of fall detection.

Mohammed Farsi [24] presented an ensemble of RNN models to detect falls in the Internet of Things (IoT) environment. The approach used variants of LSTM and ensemble methods such as bagging, stacking, Adaboost, XGBoost, and Random Forest, respectively. In this approach, random forest exhibited better results than the LSTM model. The ensemble didn’t receive the full data and it resulted in insufficient training of the data, which resulted in better performance in random forest than the ensemble of LSTM.

Khraief et al. [1] presented a multi-stream weighted deep convolutional network (MSWDCnet) to detect elderly falls. Four CNN streams were used in the approach. The RGB and depth images were combined and used in the first model. The second one concentrated on human shape variations. In order to extract the human silhouette, the background subtraction and the recognition of the person are handled. The last two models deals distinguish the movement. However, the complex background affects the features it extracts, leaving a lack of discriminant information.

Khan et al. [25] presented an approach that doesn’t depend on the training data to learn the fall types. The adversial learning approach, which consists of a spatio-temporal auto encoder (3DAE), reconstructs the videos and a spatio-temporal convolution network distinguishes the original videos. The 3D convolution learns the spatial and temporal features from the input videos. Adversarial learning detects the unseen falls with this approach. Tahir et al. [26] presented an automatic FDS where a random vector functional link (RVFL) stacking ensemble classifier is used. The Hurst exponent is used to classify the fall. The approach achieved better speed than individual classifiers. It has the lowest runtime cost and has an accuracy that is greater than or equivalent to most recent ensemble methods. The speedup benefit of the RVFL ensemble can lead to real-time implementation on limited processors.

Ding et al. [27] proposed a WiFi-based fall identification system with RNN. With discrete wavelet transform (DWT), noise is removed from the data. The RNN is used to classify ADL falls. The approach uses a Web application programming interface to provide communication. The FDS model is implemented in a mobile APP with identity authentication and other user-enabled web facilities. It has the ability to detect falls from other motions and accurately determine whether older people are experiencing falls or not. It is also very robust to the random noise produced in its interior surroundings.

Maitre et al. [28] presented a FDS with ultra wideband radars (UWR) and a deep neural network (DNN). The DNN consists of CNN stacked with LSTM and a fully connected neural network (FCNN) to recognise the fall from ADL. The richness of information produced by this method based on UWB radars is not as easily identifiable as that of pictures from cameras in terms of identifying the person or their actions. This method also learns time-series patterns and extracts features for classification better than an LSTM.

By viewing the detection as a multi-class issue, Wang et al. [37] introduced an array of sensor-based FDS and a Multi-source CNN Ensemble (MCNNE) structure to more efficiently extract the feature from multi-sensor data. In the system under consideration, data from various sensors is individually preprocessed and structured as the training dataset, and the output feature maps from various sensors are concatenated to create a total feature map. It enhances algorithm robustness while also increasing prediction accuracy.

Divya and Sri [38] introduced FDS with the use of a 3-layer (Edge-Fog-Cloud) architecture that makes use of the available smart devices. The edge detection process uses a smart device built utilising transfer learning and a compressed neural network operating on it for vision-based detection. Using sensor-based data and edge decisions, an ensemble learning approach is used to make decisions in fog. Model construction and permanent storage are both done on the cloud. This approach also makes use of augmentation for data set construction to enhance the model’s performance. Additionally, ensemble learning exhibits superior performance to decisions made by a single algorithm. The fog layer is where the collective voting (decision-making) takes place, which lowers the latency in decision delivery. Given its capacity for calculation and communication, the fog layer can also be employed to transmit alert signals.

Al-Rakhami et al. [39] introduced an efficient FDS framework based on deep learning and edge computing within 5G networks. Also, a deep gated recurrent unit (DGRU) network has been introduced to boost the effectiveness of deep learning based FDS. Since DGRU works with time-series IoT data, it offers the advantage of reducing the number of parameters and avoiding the vanishing gradient issue.

Berlin and John [40] proposed Effective Deep Learning-based Siamese (EDLS) frameworks for human FDS. Two distinct Siamese-based frameworks have been developed, one with a depth-wise separable convolutional filter and the other with a regular 2D filter. Additionally, either RGB characteristics or optical fog features are given to these frameworks. Also, the Siamese configuration, which uses optical-based features, performs better than RGB features because optical-based features are more effective at representing motion information. Despite having just a small number of samples per class label during training, the Siamese network has a strong ability for generalisation. Further Table 1 highlights the advantage and disadvantage of existing methods.

**Table 1**: Advantages and challenges of existing approaches

|  |  |  |  |
| --- | --- | --- | --- |
| **Reference** | **Technique proposed** | **Advantages** | **Challenges** |
| [20] | MTHCAE | Enhance the accuracy of fall detection by increasing the feature's representativeness using auxiliary task. | It require large dataset to achieve good performance. |
| [21] | DSTCAE | Extracts both spatial and temporal features. | It is affected by the noise in the frames. |
| [9] | fused saliency maps-based FDS | It lessens the impact of the complex background in feature extraction. | Takes large amount of time for classification. |
| [22] | FIEL | Sends alerts through IoT protocol with details on the kind of fall and where it occurred.  The most effective approach is selected by examining the classifiers' performances, computing needs, and areas under ROC curves. | It needs a large quantity of data and processing power. |
| [8] | S3D-CNN | It accurately extracts human movement features. | Processing speed is low. |
| [23] | DBN | Detect nine types of falls using forty three different features | Complex structures lengthens the training and testing periods. |
| [2] | 2C3DCAE | Minimal latency | Overfitting problem arises. |
| [18] | 3-layer fog-cloud architecture | Reduces additional network congestion delay and network latency. | Memory usage is high. |
| [24] | Ensemble of RNN model | Avoid overfitting issue. | Class imbalance problem occurs. |
| [1] | MSWDCnet | Detection accuracy is improved by integrated complimentary information such as RGB, motion, shape, and depth. | Complex structure affect the feature it extracts. |
| [25] | 3DAE | Identify falls without their training data | Noise sensitivity. |
| [26] | RVFL | It achieves fast training and minimal latency. | Large storage space is required. |
| [27] | RNN-based WiFi FDS | Data details are preserved while noise in the obtained data is removed. | Overfitting occurs. |
| [28] | FDS with UWR and DNN | It extracts features for classification better than an LSTM and learns time-series patterns. | Fall detection rate is low. |
| [29] | MCNNE | It distinguishes the fall direction and increases the detecting accuracy. | Processing time is high. |
| [30] | FDS using 3-layer (Edge-Fog-Cloud) architecture. | It lowers the latency in decision delivery. | Complex structure. |
| [31] | DGRU | It reduces the parameters and solves the vanishing gradient problem. | Imbalanced class problem. |
| [32] | EDLS | It has the capability to learn even with a small training set. | Training time is higher. |

**3. Proposed Method**

The proposed fall detection approach utilises ensemble learning where multiple ML models are used. Initially, the video is pre-processed and its features are extracted. Then the features are trained with the classifiers in the ensemble model. The ensemble model follows a majority voting approach in choosing the optimal classifiers. The majority vote employs greedy based forward, backward, and recovery search in order to choose the optimal classifiers. The proposed ensemble based FDS focuses on obtaining the optimal classifier without losing the optimal classifiers during the greedy based search with the selection criteria. Fig 1 presents the block diagram of the proposed ensemble based FDS.



**Figure 1:** The functional diagram of the proposed FDS

**3.1 Pre-processing**

The same object detection steps used in [41] are used in this approach. The pre-processing step consists of the conversion of the input frame into a grayscale image, the removal of background and foreground segmentation, noise removal, binarization, and morphological operations.

Finding a scene without a human in the detecting zone is called the initial background. For foreground segmentation, the initial background and the pictures from the video streams will first be transformed to grayscale images. The frame will then be compared against the background to look for any changes using Gaussian background models (GMM).The equation (1) determines the likelihood that a pixel of a specific distribution will occur at time . On the basis of least variance and highest weight, the distributions are arranged.

 (1) where represents Gaussian distributions number,  represents the  Gaussian’s weight estimate in the mixture at that time,  represents the probability density Gaussian function  represent  represents mean value,  represents the  Gaussian’s covariance matrix at time . The outcome of GMM operation is then sent to a shadow removal technique, which finds and eliminates any foreground shadows.

The image will next go through a process called binarization to convert it from grayscale to binary (black and white). Binary images are created when pixels with lightness below the filter threshold () acquire the low colour (0), while pixels with lightness above the acquire the high colour (1). The object’s shadow and light illumination must also be taken into account. Higher  will distort the shape of the human body while lower  will increase the impact of shadow in the image and increase pixel noise. Therefore, the  value chosen should reduce both impacts. The best T value, according to numerous tests, is 15. After the shadows were removed, some foreground pixels were mistaken for shadows and were thus removed. There is some slight deformation in object shape, so morphological operations are used to reconstruct the deformation.

**3.2 Feature extraction**

There are a variety of strategies for extracting information from fall detection systems depending on the kind of data. In vision-based approaches, optical flow algorithms (OFA) [28] give important essential regarding motions in pictures. So in this approach OFA [1] is used to extract the features. The OFA provides the information about the movements in the image without considering the static characteristic in the image. This technique determines the fall event's magnitude, direction, and movement. In fact, Histograms of Optical Flow Orientation and Magnitude (HOFM), and other hand-crafted features routinely employ this data to define motion information. Different fall detection techniques rely less on optical flow displacement and more on the magnitude and direction as hand-crafted characteristics.

To extract deep motion information, this method utilise the size and direction of optical flow. In order to include this information in the temporal stream, the optical flow is computed first, after that the magnitude  and orientation  information are computed by eqn. (2 &3).

 (2)

 (3)

where  represents the pixels relative vertical motion and  represents the pixels relative horizontal motion on the images which are used to compute optical flow. These in-depth characteristics were fed into an ensemble classifier that categorises events as “fall” or “no fall” depending on their input.

**3.3 Models used in the Ensemble Approach**

In this section, the ensemble learning models are described. In the proposed fall detection system, four different classifiers are selected for the classification task.

***SVM Classifier***

The *support vector machine* (SVM) is a supervised machine learning approach which uses a hyperplane in order to maximize the margin. The SVM function is expressed in equation (4).

 (4)

Where  is the mapping of a sample *x* from the input space to a higher dimensional feature space, *w* is the coefficient vector, and *b* is the offset of the hyper plane from the origin. The SVM proposed in our research uses the Radial Basis Function (RBF) [30].

***KNN Classifier***

The KNN classification predicts the test sample from the training sample through a distance measure. The KNN classifier classifies the activities of a person based on their features. The Euclidean distance is used to find the distance between objects. The K value has to be chosen carefully since the larger K values can create gaps between the classes. Cross validation is used to select the best of the class [31].

***Decision tree algorithm***

The decision tree algorithm is used for classification and has been used in many problems, including FDS. The decision trees (DT) are made up of nodes and branches, which have a decision making model with a tree graph. In DT, there are a lot of paths or branch nodes, which help to take decisions based on the leaf nodes [21].

***Deep-LSTM (D-LSTM) Model***

The LSTM is an extension of the RNN. The structure of D-LSTM is shown in Figure 2. The LSTM has an input gate, a forget gate, and an output gate. These gates control the state of the cell as well as can change their own state. The Deep LSTM is considered to work better than conventional LSTM for complex tasks. The classification performance can be increased with a greater number of layers. Deep architecture is better at learning complex data [32], [33].In this model, the D-LSTM is utilized to predict temporal memory and visual attention. The TSP equations are given below.



**Figure 2:** Structure of D-LSTM model

 ʘ) (5)

 (6)

 (7)

 (8)

 ʘ ʘ  (9)

 (10)

 ʘ  (11)

 (12)

Where,  denotes the input gate,  denotes the output of cell state,  denotes the input block,  denotes the input of cell state, denotes the forget gate, is an output gate, is an hidden block, P represents the bias vectors,  denotes the input vector at time F, the *v* denotes the rectangular input weighted matrices, ʘ denotes the scalar product of vector and  denotes the average logistics sigmoid function.  represent the weight matrices are connected to the of cell input and three gates.  denotes the weight matrices are connected to the  of three gates and cell state, The output of this system is denoted by and  is a previous hidden state.

**3.4 Majority voting with greedy algorithm**

 In an ensemble classifier, various individual classifiers are combined for the classification purpose. The accuracy and diversity properties of the classifiers are utilised in the ensemble approach to obtain better results. The outcomes of the diverse approaches are different from each other, and in an ensemble the outputs are combined to obtain better classifiers. The ensemble is modelled at various levels, namely combination, classifier feature, and data level. The proposed ensemble approach follows the combination level of various classifiers. The voting approach is one of the ensemble approaches. The voting approaches generally used are majority voting, weighted voting, average and product of probabilities, and minimum and maximum probabilities [16], [19]. The proposed voting is carried out based on a greedy search approach which identifies the potential classifiers which can accurately distinguish fall from non-fall. The quality of the ensemble system depends on the selection criteria [34]. The selection criteria on majority voting with greedy algorithm is illustrated in Figure 3.

*Selection criteria*

The proposed voting approach uses majority voting error, goodness of a classifier, majority voting improvement (MVI), and N best classifier performance as the selection criterion. The MVI is a measure which gives the improvement of the ensemble combined performance when compared to individual classifier performance [35]. It is given as in equation (13).

*MVI=MVE-EMER* (13)

The ensemble consists of N classifiers, Let be the input samples and  be the output of the *jth* classifiers for *i* input samples. The input samples are given as , where the values of s and t are output of the classifiers,

The error of the classifiers used in ensemble for the input sample  is given as in equation (14).

 (14)

Where  is the classifier output binary output of the jth classifier for the *i* th input sample. The error rate of the *j*th classifier is given as in equation (15).

 (15)

The ensemble mean error rate (EMER) is given as in equation (16).

 (16)

The majority voting error (MVE) rate is given as in equation (17).

 (17)

Where  (18)

The goodness of the classifier is chosen as the time consumed by a classifier to complete the classification process at its maximum iteration. The greedy algorithm that our proposed approach employs performs forward, backward, and recovery search. The forward search (FS) is a greedy algorithm in which at every iteration an individual classifier which can reduce the MVE is checked. Each cycle begins with the best single classifier and then seeks a pair of classifiers that minimises the MVE. Along with the MVE, the goodness of the classifier is also evaluated. In the FS search, the addition of the classifier takes place. If MVE is not decreased for any classifier pair, the algorithm terminates with the currently generated combination. Finally, it adds the classifiers with low MVE and low time to perform the classification process.

A classifier selection strategy that is symmetrical to FS greedy is backward search. This method starts with the whole ensemble of classifiers, and during each iteration, it seeks a pair of classifiers from the combination that, if removed, would significantly enhance the performance of the classifier or any other criterion used in the selection process. Here, in backward search, the pairs of classifiers are searched based on the selection criteria called majority voting improvement and the classifiers are removed based on the criteria with the objective of maximising the ensemble performance.

**Figure 3:** Application of the selection criteria on majority voting with greedy algorithm

When some pairs of classifiers are removed in the BS stage, there may be chances for the loss of some classifiers which can perform better. The stochastic nature of the ML algorithms may cause different results at each time. Even small decision changes used in ML algorithms can cause a lot of variations in the output. Therefore, once the BS ends, a recovery search is initiated to preserve the loss of the classifiers. A recovered search checks the removed classifiers by the BS with the N best classifiers approach. For this purpose, the removed classifiers are sorted and each one's individual performance is analysed for maximum iterations, and the best of the classifiers is chosen. Once the search process ends, the optimal classifiers are fused and the final classification is carried out.

**4. Experimental results**

The simulations of the proposed system are carried out in Python 3.9.0. The program was conducted with an Intel Core i7 processor and 4 GB RAM.

**4.1 Dataset description**

The video dataset used in the proposed approach is collected from [36]. With a single camera, the video surveillance of the elderly at home as well as the office room is captured. The resolution is 320\*240 pixels and the frame rate is 25 frames per second. The video sequence has problems like illumination, occlusion, and texture background. The dataset consists of both falls and ADLs captured by the camera from various locations. The dataset consists of 191 videos, which are annotated along with ground truth of the fall position. The video was shot in a variety of settings, including the author's home, a coffee shop, and a lecture hall. The dataset for fall detection is not limited to the same location. The dataset consists of a sequence of falls while walking, and daily activities that include walking, bending, sitting etc. The training and testing set is adjusted to a ratio of 4:1. The input image from the dataset is given in figure 4.

|  |  |
| --- | --- |
| D:\Sikha-ii\1.jpg  (a) | D:\Sikha-ii\2.jpg  (b) |
| D:\Sikha-ii\3.jpg  (c) | D:\Sikha-ii\4.jpg  (d) |
| D:\Sikha-ii\5.jpg  (e) | |

**Figure 4:** Frames from the dataset illustrating a) shows the person who is lying on floor after a fall, b) sitting on a sofa c) standing posture d) bending towards the couch, e) sleeping on the couch.

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | | |
| 1. Manual Annotation | | |
|  | |  |
|  | |  |
| |  | | --- | | (b) Abnormal Annotation |   **Figure 5:** Dataset (a) manual Annotation and (b) Abnormal Annotation | | |

Figure 5 illustrates the dataset detection of manual and abnormal annotation. Here, the manual annotations are detected in the green frame and the abnormal annotations are detected in the red frame.

***4.2* Performance metrics evaluation**

The results obtained from the simulation are analysed with metrics like accuracy, sensitivity, specificity, and precision. Accuracy (ACC) provides the correct difference between the fall and non-fall. It provides the correct predicted samples.

 (19)

Sensitivity signifies the capacity of the model to identify the falls correctly without identifying the ADLs like walking or sitting, which are non falls.

 (20)

Specificity is the capacity of the model to detect falls. Non-falls like walking, climbing, or lying shouldn’t be identified as falls.

 (21)

TP is the success of the ensemble model in identifying a fall or ADL when it happens. TN depicts the success of the model in identifying the absence of a fall or non-fall when it doesn’t occur. FP is the failure in detecting the fall or non-fall when it doesn’t happen, and FN is the failure of the ensemble model in identifying the fall or non-fall when it happens.

|  |  |
| --- | --- |
|  |  |
| (a) SVM | (b) KNN |
|  |  |
| (c) Decision tree | (d) Deep LSTM |
|  | |
| (e) Proposed model | |

**Figure 6:** The confusion matrix of fall detection for ensemble classifiers

Figure 6 illustrates the confusion matrices for different classifiers. The performance of the classifier is evaluated by using 10 fold cross validation. Here, normal and abnormal fall detection are represented by TP (orange), TN (blue), FP (white) and FN (gray). Table 2 provides the performance comparison between the individual classifiers used in the ensemble model and the ensemble model.

**Table 2:** Comparison of individual classifier performance with ensemble model

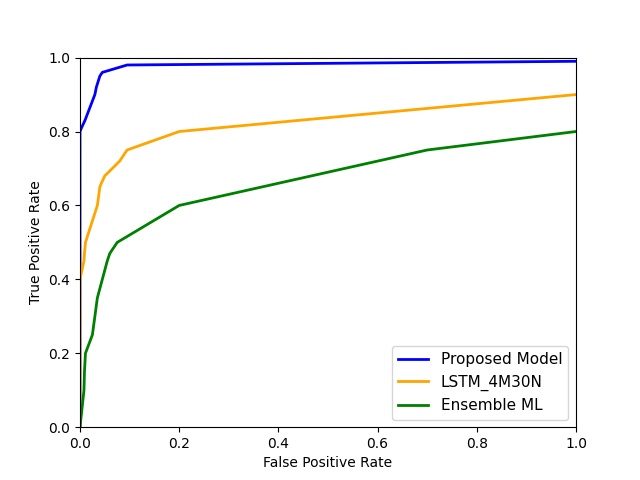
|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Sensitivity** | **Specificity** |
| SVM  kNN  Decision Tree  Deep LSTM  **Proposed model** | 78.34  81.23  93.91  94.35  **99.98** | 77.23  80.32  92.12  93.43  **99.80** | 76.45  79.34  91.2  92.20  **99.90** |

The proposed ensemble model has been able to exhibit better performance than the individual models. The ensemble model has been successful in selecting the optimal classifiers, which has resulted in better performance in terms of accuracy, sensitivity, specificity, and precision. Table 3 provides the performance comparison of the proposed ensemble with the conventional approaches.

**Table 3: Performance comparison of proposed ensemble with conventional approaches**

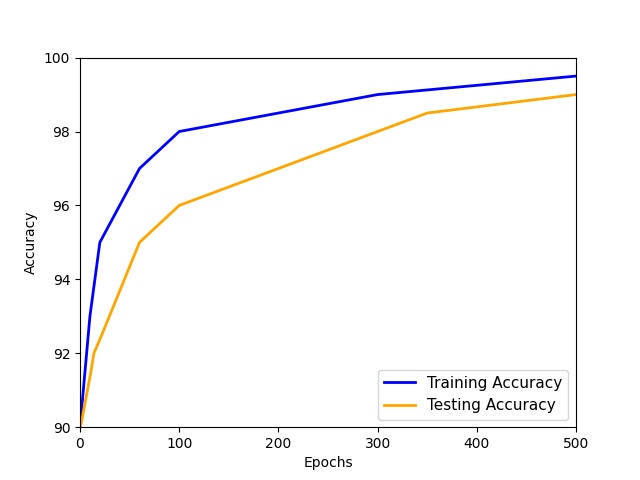
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Sensitivity** | **Specificity** | **Precision** |
| 2D CNN [15] | - | 97.95 | 83.08 | - |
| Multi-task hourglass convolutional  auto-encoder [20] | 0.962 | 1 | 0.930 | 0.92 |
| Multi-Stream CNN [1] | 99.72 | 99.70 | 99.80 | - |
| LSTM\_4m30N [24] | 0.934715 | - | - | 0.920241 |
| Ensemble ML Algorithm [21] | 98.72 | 96.22% | 94.60% | - |
| **Proposed model** | **99.98** | **99.80** | **99.90** | **96.23** |

It is observed that the conventional approaches have failed to exhibit a better performance than the conventional schemes. This shows that the optimal classifier selection in the ensemble model has achieved better performance.



**Figure 7:** Region of operating characteristics

Figure 7 presents the ROC curve comparison of the proposed approach with conventional schemes for false positive and true positive ratio. The proposed ensemble model is denoted by the blue line, which has better ROC performance when compared to the conventional ensemble models due to its higher classification accuracy.



**Figure 8:** Accuracy of Ensemble over different epochs

Figure 8 presents the accuracy of the proposed ensemble approach for different epochs. The proposed ensemble has an epoch of 500. It is observed that with increase in epochs the accuracy of the model increases.

D:\Sikha-ii\RESULTS\Figure_1.tiff

**Figure 9:** Loss of the proposed ensemble model

Figure 9 presents the loss of the proposed ensemble approach with increasing epochs. With an increasing number of epochs, the proposed ensemble displayed a reduction in loss. The performance is tested for both the training and testing stages.

**4.3 Computation complexity evaluation**

Table 4 shows the computational complexity (CC) of the proposed ensemble and individual classifiers. In SVM, CC is, where  represents the support vectors and  represents the features extracted. Its CC is low, but its classification accuracy is also very low. The CC of KNN is, where represents the number of training vectors and  represents the number of nearest neighbours. Computational complexity of Decision Tree , where  represents number of training examples. The proposed ensemble model computational complexity iswhere  represents the size of the dataset, represents the original size of the ensemble and denotes the computational complexity of the evaluation criteria. The proposed ensemble model’s complexity is minimised by applying the simplest selection criteria and search techniques. It not only reduce complexity it also improves the performance by increasing classification accuracy.

**Table 4: Computational complexity comparison of proposed ensemble with individual classifier**

|  |  |  |
| --- | --- | --- |
| **Methods** | **Complexity** | **Accuracy** |
| SVM |  | 78.34 |
| KNN |  | 81.23 |
| Decision Tree |  | 93.91 |
| Deep LSTM |  | 94.35 |
| **Proposed model** |  | **99.98** |

**5 Conclusion**

In this article, a novel ensemble-based FDS with a greedy algorithm is designed for accurate fall detection. After pre-processing and feature extraction, the optimal classifiers are selected using greedy based majority voting: forward, backward, and recovery search. This search-based approach chooses the optimal classifiers with selection criteria (MVE, goodness of a classifier, MVI, and N best classifier performance) that help to prevent the loss of optimal classifiers during the search process. During the searching process, the forward search adds the classifiers that complete the classification process in a shorter amount of time, while the backward search removes the low- performance classifier. Also, loss of best classifiers during backward search is avoided using recovery search. The experimental result is analysed in terms of accuracy, sensitivity, specificity, and precision. It exhibited better results than the conventional approaches through its search-based selection of classifiers. In future work, we will develop this proposed model with the transfer learning approach to detect unseen falls in various camera positions and other health-related abnormal behaviour.

**Compliance with Ethical Standards:**

**Funding:** There is no funding for this study.

**Conflict of Interest:**  Authors declares that they have no conflict of interest.

**Ethical approval:** This article does not contain any studies with human participants and/or animals performed by any of the authors.

**Informed consent:** There is no informed consent for this study.

**Authors' contributions**

All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

**Data Availability Statement:**

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Shikha Rastogi, Jaspreet Singh. The first draft of the manuscript was written by Shikha Rastogi and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Conceptualization: Shikha Rastogi; Methodology: Shikha Rastogi; Formal analysis and investigation: Shikha Rastogi, Jaspreet Singh; Writing - original draft preparation: Shikha Rastogi; Writing - review and editing: Jaspreet Singh; Supervision: Jaspreet Singh.

**Reference**

1. Khraief C, Benzarti F, & Amiri H (2020). Elderly fall detection based on multi-stream deep convolutional networks. Multimedia Tools & Applications, 79.
2. Mehta V, Dhall A, Pal S, & Khan S (2020). Motion and region aware adversarial learning for fall detection with thermal imaging. arXiv preprint arXiv:2004.08352.
3. Geertsema EE, Visser GH, Viergever MA, & Kalitzin SN (2019). Automated remote fall detection using impact features from video and audio. Journal of biomechanics, 88:25-32.
4. Falls in Older Persons: Risk Factors and Prevention. Accessed: Oct. 14, 2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK235613/>.
5. Islam MM, Tayan O, Islam MR, Islam MS, Nooruddin S, Kabir MN, & Islam MR (2020). Deep Learning Based Systems Developed for Fall Detection: A Review. IEEE Access, 8:166117-166137.
6. Lin CB, Dong Z, Kuaz WK, & Huang YF (2021). A Framework for Fall Detection Based on OpenPose Skeleton and LSTM/GRU Models. Applied Sciences, 11(1):329.
7. Ambrose AF, Paul G, & Hausdorff JM (2013). Risk factors for falls among older adults: a review of the literature. Maturitas, 75(1):51-61.
8. Xiong X, Min W, Zheng WS, Liao P, Yang H, & Wang S (2020). S3D-CNN: skeleton-based 3D consecutive-low-pooling neural network for fall detection. Applied Intelligence, 50(10):3521-3534.
9. Li H, Li C, & Ding Y (2021). Fall detection based on fused saliency maps. Multimedia Tools and Applications, 80(2):1883-1900.
10. Xu T, Zhou Y, & Zhu J (2018). New advances and challenges of fall detection systems: A survey. Applied Sciences, 8(3):418.
11. Cippitelli E, Fioranelli F, Gambi E, & Spinsante S (2017). Radar and RGB-depth sensors for fall detection: A review. IEEE Sensors Journal, 17(12):3585-3604.
12. Mubashir M, Shao L, & Seed L (2013). A survey on fall detection: Principles and approaches. Neurocomputing, 100:144-152.
13. Gutiérrez J, Rodríguez V, & Martin S (2021). Comprehensive review of vision-based fall detection systems. Sensors, 21(3):947.
14. Islam MM, Tayan O, Islam MR, Islam MS, Nooruddin S, Kabir MN, & Islam MR (2020). Deep Learning Based Systems Developed for Fall Detection: A Review. IEEE Access, 8:166117-166137.
15. Espinosa R, Ponce H, Gutiérrez S, Martínez-Villaseñor L, Brieva J, & Moya-Albor E (2019). A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset. Computers in biology and medicine, 115:103520.
16. Maanijou R, & Mirroshandel SA (2019). Introducing an expert system for prediction of soccer player ranking using ensemble learning. Neural Computing and Applications, 31(12):9157-9174.
17. Bulotta S, Mahmoud H, Masulli F, Palummeri E, & Rovetta S (2013). Fall detection using an ensemble of learning machines. In Neural Nets and Surroundings. Springer, Berlin, Heidelberg, 81-90.
18. Sarabia-Jácome D, Usach R, Palau CE, & Esteve M (2020). Highly-efficient fog-based deep learning AAL fall detection system. Internet of Things, 11:100185.
19. Savargiv M, Masoumi B, & Keyvanpour MR (2020). A new ensemble learning method based on learning automata. Journal of Ambient Intelligence and Humanized Computing, 1-16.
20. Cai X, Li S, Liu X, & Han G (2020). Vision-based fall detection with multi-task hourglass convolutional auto-encoder. IEEE Access, 8:44493-44502.
21. Nogas J, Khan SS, & Mihailidis A (2020). Deepfall: non-invasive fall detection with deep spatio-temporal convolutional autoencoders. Journal of Healthcare Informatics Research, 4(1):50-70.
22. Yacchirema D, de Puga JS, Palau C, & Esteve M (2019). Fall detection system for elderly people using IoT and ensemble machine learning algorithm. Personal and Ubiquitous Computing, 23(5):801-817.
23. Jahanjoo A, Naderan M, & Rashti MJ (2020). Detection and multi-class classification of falling in elderly people by deep belief network algorithms. Journal of Ambient Intelligence and Humanized Computing, 1-21.
24. Farsi M (2021). Application of ensemble RNN deep neural network to the fall detection through IoT environment. Alexandria Engineering Journal, 60(1):199-211.
25. Khan SS, Nogas J, & Mihailidis A (2021). Spatio-temporal adversarial learning for detecting unseen falls. Pattern Analysis and Applications, 24(1):381-391.
26. Tahir A, Morison G, Skelton DA, & Gibson RM (2020). A novel functional link network stacking ensemble with fractal features for multichannel fall detection. Cognitive Computation, 12(5), 1024-1042.
27. Ding J, & Wang Y (2020). A WiFi-based smart home fall detection system using recurrent neural network. IEEE Transactions on Consumer Electronics, 66(4):308-317.
28. Maitre J, Bouchard K, & Gaboury S (2020). Fall detection with UWB radars and CNN-LSTM architecture. IEEE journal of biomedical and health informatics, 25(4):1273-1283.
29. Cai X, Li S, Liu X, Han G (2019). A novel method based on optical flow combining with wide residual network for fall detection. In2019 IEEE 19th International Conference on Communication Technology (ICCT) IEEE, 715-718.
30. Dou J, Yunus AP, Bui DT, Merghadi A, Sahana M, Zhu Z, & Pham BT (2020). Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan. Landslides, 17(3):641-658.
31. Liu CL, Lee CH, & Lin PM (2010). A fall detection system using k-nearest neighbor classifier. Expert systems with applications, 37(10):7174-7181.
32. .Nagabushanam P, George ST, & Radha S (2019). EEG signal classification using LSTM and improved neural network algorithms. Soft Computing, 1-23.
33. Tian Y, Zhang K, Li J, Lin X, & Yang B (2018). LSTM-based traffic flow prediction with missing data. Neurocomputing, 318:297-305.
34. Ulaş A, Semerci M, Yıldız OT, & Alpaydın E (2009). Incremental construction of classifier and discriminant ensembles. Information Sciences, 179(9):1298-1318.
35. Ruta D, & Gabrys B (2005). Classifier selection for majority voting. Information fusion, 6(1), 63-81.
36. Dataset link: <https://imvia.u-bourgogne.fr/en/database/fall-detection-dataset-2.html>
37. Wang L, Peng M, Zhou Q (2020). Pre-impact fall detection based on multi-source CNN ensemble. IEEE Sensors Journal. 20(10):5442-51.
38. Divya V, Sri RL (2020). Docker-based intelligent fall detection using edge-fog cloud infrastructure. IEEE Internet of Things Journal. 8(10):8133-44.
39. Al-Rakhami MS, Gumaei A, Altaf M, Hassan MM, Alkhamees BF, Muhammad K, Fortino G (2021). FallDeF5: A Fall Detection Framework Using 5G-Based Deep Gated Recurrent Unit Networks. IEEE Access. 9:94299-308.
40. Berlin SJ, John M (2021). Vision based human fall detection with Siamese convolutional neural networks. Journal of Ambient Intelligence and Humanized Computing. 1-2.
41. Hafiz F, Shafie AA, Khalifa O, Ali MH (2010). Foreground segmentation-based human detection with shadow removal. InInternational Conference on Computer and Communication Engineering (ICCCE'10) IEEE, 1-6.