Journal of Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.777981

HYBRID DEEP LEARNING APPROACH FOR HUMAN FALL DETECTION

Bhavik Jain, Dr. Naveen Choudhary

1Department of Computer Science and Engineering, College of Technology and Engineering (CTAE),

Maharana Pratap University of Agriculture and Technology, Udaipur (Rajasthan)

ABSTRACT

Falls are a significant contributor to unintentional fatalities and injuries on a global scale. People with balance impairments are particularly susceptible to falling. While surveys and lab simulations have provided valuable insights, there is a lack of prospective data on actual falls in real-world settings. Acquiring such information is crucial for identifying fall risks and designing effective systems for detecting and alerting falls. The technological areas of machine vision and ML are based on the ability of an individual to trace the actions of others. Action is a term used to describe a sequence of bodily functions that include several body parts working together at once. The comparison is conducted on any kind of remark against a predetermined pattern in machine vision, and the activity is recognized and labelled later on. The SVM is the classifier which is applied in the previous work for recognizing the activities of individuals. The SVM classifier performs poorly for identifying human actions. The presented work suggests a hybrid approach which is the combination of CNN and LSTM model. The proposed model achieves an accuracy of upto 87 percent for the human fall detection.

Keywords: Fall Detection, SVM, CNN, LSTM, Sensors

1. INTRODUCTION

Falls are a major concern for the elderly people, resulting in injuries and hospitalizations. The annual incidence of falls among individuals of 75 and above is estimated to be at least 30%, whether living independently or in institutions [1], [2]. According to the Centers for Disease Control and Prevention, approximately 2.8 million older individuals are treated in emergency departments for fall-related injuries, each year. One out of five falls leads to severe consequences such as fractures or head injuries. The severity of the outcome largely depends on how long the person remains un-assisted after the fall.

Traditional solutions involve Personal Emergency Response Systems (PERS), which are compact and lightweight, and battery-powered devices equipped with a distress button and provided to the elder people. However, these systems are relied on the button that should be pressed for assistance, but it may not be possible if the person becomes unconscious. Therefore, it is crucial to develop a system that can automatically detect falls and initiate a request for help.

Smartphones have become incredibly common in today's society, and this includes a large number of elderly individuals as well [3], [4]. By integrating a fall detection system into smartphones, there is no longer need to purchase and carry an additional device like a Personal Emergency Response System (PERS). With smartphones, an automatic messages can be sent to the emergency contacts when a fall is detected. Most of the smartphones are equipped with various built-in sensors that can measure motion, orientation, and environmental conditions. These sensors provide precise and accurate raw data, which makes them suitable for monitoring device movement, positioning, and changes in the surrounding environment. With the widespread use of smartphones being an essential part of people's lives, incorporating a sensor-based fall detection system can significantly enhance their contribution to healthcare support [5], [6]. Figure 1 illustrates a typical fall detection system based on smartphone sensors.

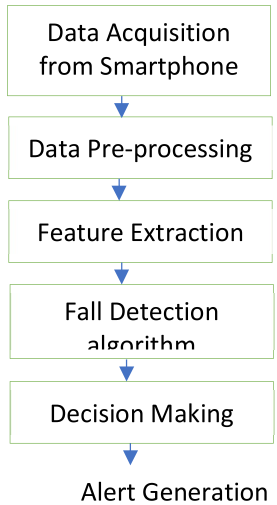

Figure 1. Pipeline of Fall Detection based on smartphone sensor data

A general fall detection system based on the sensors of a smartphone is typically consisted of the following stages:

i. Sensor Data Acquisition: The smartphone's built-in sensors, such as the accelerometer and gyroscope, are used to capture motion and orientation data. These sensors provide information about the smartphone's movement.

ii. Data Preprocessing: The acquired sensor data is pre-processed to remove noise, filter out irrelevant information, and normalize the data. Preprocessing techniques may include filtering, smoothing, and feature extraction to prepare the data for further analysis [7], [8].

iii. Feature Extraction: Relevant features are extracted from the pre-processed sensor data. These features are used to capture key characteristics of the user's motion patterns and can be used to distinguish between normal activities and fall events. Commonly used features include acceleration magnitude, orientation angles, and changes in velocity.

iv. Fall Detection Algorithm: The extracted features from the fall detection system are fed into a fall detection algorithm to determine if a fall event has occurred. Machine learning techniques, including threshold-based methods, rule-based approaches, and supervised learning algorithms, can be employed to classify fall events based on these features. Some commonly used machine learning algorithms are:

- Support Vector Machine (SVM): The SVM is a supervised machine learning model that aims to find a hyperparameter in an n-dimensional space to separate data points of different classes. There are two types of SVM: linear and non-linear [9], [10]. The linear classifier assumes linear separability of data points and identifies the optimal hyperplane of maximum margin among classes. On the other hand, the non-linear classifier maps the data points onto a higher-dimensional space using a kernel and finds a discriminant function associated with the hyperplane in the transformed space. In SVM, the data points are plotted in a plane, and the hyperplane acts as a decision boundary to classify the data points from different classes. Support vectors are the data points located near the hyperplane and play a crucial role in determining the position of the hyperplane. The SVM seeks to generate a decision boundary with the maximum gap between classes, making it a useful classifier for distinguishing fall activities from regular daily activities [11], [12]. For instance, in Figure 3, class 1 represents the pattern of regular walking, while class 2 indicates the features associated with a fall event. The SVM can effectively differentiate between these two classes based on the extracted features.

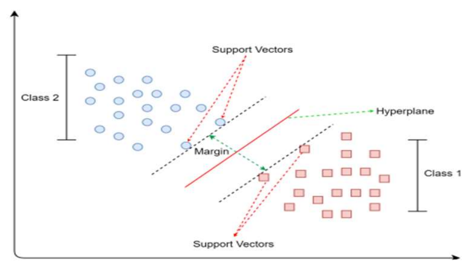

Figure 2. Support vector machine algorithm

- Random Forest (RF): Random Forest is a supervised classification algorithm that consists of multiple decision trees. Each decision tree uses test nodes and branches to form a classification based on the data. However, decision trees are sensitive towards the changes in the training data, which can affect the overall tree structure. To address this, Random Forest uses a technique called bagging, where each tree randomly selects a subset of the data to create diverse decision trees. The Random Forest combines the predictions of these trees through voting, where the majority class is considered as the final prediction. This approach helps to overcome issue of overfitting and is useful for classifying Activities of Daily Living (ADL) from fall activities in fall detection systems [13], [14].
- k-Nearest Neighbors (kNN): The k-Nearest Neighbours (kNN) algorithm is a supervised learning approach used for classification and regression tasks. It is relied on the assumption that similar data points are located close to each other. To determine proximity, it calculates the Euclidean distance between data points. The kNN algorithm is used for selecting a value for k, which represents the number of neighbours to consider each data point. By finding the nearest neighbours based on distance, K-NN establishes the boundaries for clusters or classes. The algorithm creates a sorted list of distances between each point and other points, and the top k entries are used for classification or regression [15], [16]. One advantage of K-NN is its simplicity and low computational cost, as it does not require additional assumptions. Due to its feasibility, it finds applications in recommendation systems. For fall detection, K-NN can be utilized to differentiate between fall behaviours and non-fall behaviours.

- k-Means: The k-means algorithm is an unsupervised machine learning technique that is both simple and extensive. Its major purpose is to group similar data points together in order to uncover underlying patterns. By defining the number of clusters, denoted with "k", this algorithm identifies distinct clusters within a dataset for classification purposes. Each cluster represents a set of data points that share similarities, such as specific activities like gait or fall patterns. It determines the optimal number of centroids, which serve as the center points for each cluster [17], [18]. Data points are then assigned to the nearest cluster based on proximity to the centroids, with the goal of minimizing the size of the centroids. This technique is commonly applied in data cluster analysis and feature learning tasks. However, the performance of the k-means algorithm may vary, as even slight changes in the data can result in high variance. Consequently, it is not frequently used in fall prevention or detection algorithms.
- Linear Discriminant Analysis (LDA): LDA (Linear Discriminant Analysis) is a supervised technique that is commonly employed for dimensionality reduction [19], [20]. This algorithm serves as a pre-processing step to transform high-dimensional data into a lower-dimensional one, which can help to reduce computational costs and resource requirements. For gait analysis, where large datasets with similar patterns are often obtained using wearable devices, LDA is proved useful in reducing the dimensionality of the data. This is particularly effective method when processing is performed on low-power devices, as it enables more efficient and precise data analysis.

v. Decision Making: Once a fall event is detected, a decision-making module is executed to analyze the context and severity of the fall. This module considers the additional information such as the user's location, time of day, or medical history, to make a precise decision [21].

vi. Alert Generation: When a fall is confirmed, the system leads to generate an alert to notify relevant authorities or emergency services. This alert is send in the form of a notification, message, or automated call to selected contacts.

2. LITERATURE REVIEW

D. Pan, et.al (2021) suggested a heterogeneous sensor data fusion (HSDF) algorithm on the basis of wearable wireless body area network (WBAN) [22]. Moreover, a system was built to monitor the activities of daily living (ADLs) and fall of aged persons at higher precision and stability. Initially, three-axis acceleration, three-axis magnetic, and three-axis angular velocity sensors were selected for monitoring the activities of the elder people. Subsequently, heterogeneous sensors were exploited for transmitting the gathered data to smartphones via Bluetooth. The mobile phone network was utilized for communicating with service centers and users. The Kalman filter algorithm was implemented for mitigating the noise in system and making it more stable. The experimental outcomes indicated that the suggested algorithm yielded a sensitivity of 98.7% and specificity of 98.5%, and its practicality was proved to protect the healthy life of the elderly.

N. Biancone, et.al (2021) introduced a fall detection technique and a mobile application which was planned on the basis of sensors, placed on smartphones, such as accelerometer, gyroscope, proximity sensor, microphone and GPS [23]. This technique deployed a threshold-based algorithm for integrating the data taken from the utilized sensors. The data was pre-processed to detect fall, and a first aid request allowed to forward a message to rescuers. The thresholds and time windows were employed in this technique. The smartphone had considered its sensors for monitoring the user movements. This technique utilized various thresholds to become more effectual. The values were integrated with the streams from gyroscope and accelerometer to recognize the falls. The results exhibited the supremacy of the introduced technique over existing methods for detecting the fall.

Y. Harari, et.al (2021) projected an approach in which accelerometer and gyroscope of smartphone were exploited for monitoring the motion of users and detecting the falls based on a regularized logistic regression (RLR) [24]. A cloud server was employed for storing the data on falls and near-fall events. This approach aimed to log the fall-related variables onto a web portal which was built to discover the data such as fall probability, and location and activity of a person prior to fall. The smartphones resulted in maximizing the adaptability of this system for detecting fall. This approach was capable of detecting a fall in real time, locating the faller, and sending a notification in an automatic way. The projected approach offered a specificity of 99.9% and enhanced the precision up to 37.5% for detecting a fall.

X. Lv, et.al (2020) developed a cost effective system on the basis of You Only Look Once-version 3 with LiteFlowNet (YCL) algorithm and multi-sensor fusion (MSF) to detect and monitor the fall [25]. The initial algorithm was implemented for detecting the fall activities and the latter model considered the embedded sensors of a smartphone namely acceleration and air pressure sensors for detecting the fall actions. The developed system focused on fusing the outputs of these techniques for creating a final result with regard to accuracy of detecting fall. The confirmation of fall action allowed the smartphone to transmit an alert SMS instantly to the emergency contact for seeking medical help. According to the experimental results, the developed system was helpful for mitigating the false alarm rate (FAR), and detecting the fall in the blind zone of the camera.

T. Vaiyapuri, et.al (2021) designed an IoT based model for detecting fall using optimal deep convolutional neural network (IMEFD-ODCNN) for smart homecare [26]. This framework deployed the smartphones and an intelligent deep learning (DL) algorithm for detecting the incidence of falls in the smart home. First of all, the devices of Internet of things (IoT) were employed to capture the input video which was resized, augmented and normalized based on min-max for pre-processing the data. Moreover, the suitable feature vectors were generated to detect fall using a SqueezeNet algorithm. The Salp Swarm Optimization (SSO) algorithm was adopted to tune the hyperparameter. In the end, a sparrow search optimization algorithm with variational autoencoder (SSOA-VAE) based model was put forward to classify the fall and non-fall events. After detecting a fall, an alert was send to the caretakers and hospital management via a smartphone. UR dataset and multiple cameras fall dataset were applied for quantifying the designed model. The results revealed that the designed framework offered an accuracy of 99.76% on first dataset and 99.57% on second one.

Z. Mohammad, et.al (2023) intended a wearable monitoring model for detecting falls in which a safety technique was deployed to alleviate the fall-related injuries and send a remote notification when the body fell on the ground [27]. An ensemble deep neural network (DNN) method of Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) was suggested and analyzed. The initial algorithm was assisted in extracting the features from accelerometer and gyroscope data, and the latter one was employed for modelling the temporal dynamics of the falling procedure. A distinct class-based ensemble model was presented in which every ensemble model led to recognize a particular class. The results of SisFall dataset validated that the intended model provided a mean accuracy up to 95% to detect non-fall, 96% for pre-fall events, and 98% for detecting fall.

A. Choi, et.al (2022) established a modified directed acyclic graph-convolution neural network (DAG-CNN) algorithm with optimizing the hyperparameter for detecting fall, near-fall, and activities of daily living (ADLs) [28]. Unlike the traditional methods, this algorithm was useful for enhancing the accuracy up to 7% for detecting the fall. Moreover, this algorithm detected the near-fall events at an accuracy of 98% when the gyroscope and accelerometer attributes were integrated. The acceleration and angular velocity were integrated to make the trained model more effective. The near-falls were monitored for offering the information with the purpose of handling the risk of falls and computing the rehabilitation status of the elderly with weak balance.

D. -M. Ding, et.al (2022) devised an enhanced technique namely Precondition and Limit Threshold-Sequential Probability Ratio Test (PLT-SPRT) for detecting fall and a new system on the smart walker on the basis of the devised technique PLT-SPRT [29]. The Kalman filter model was exploited for fusing the signals of the upper and lower limb sensors. A recognizing technique was adopted for attaining the admittance control metrics. The enhanced SPRT algorithm was assisted in setting the null and the alternative assumptions, creating the probability ratio and optimizing the decision function so that the occurrence of falls was detected. The MATLAB was executed for simulating the devised approach. The embedded system was used on the basis of STM32 of the smart walker equipment for verifying and validating the efficacy of this approach for detecting the fallen state at lower delay. The devised approach offered an accuracy of 94.9% for detecting fall for smart walkers.

H. Ponce, et.al (2020) recommended a precise fall detection method after determining the least amount if sensors on the UP-Fall Detection dataset [30]. An analysis was conducted on 5 wearable sensors and 2 camera views at individual level. These sensors were integrated at feature level for computing and selecting the appropriate single or joint sources of information. The analysis indicated that a wearable sensor at the waist and a lateral perspective from a camera yielded an accuracy of 98.72%. Moreover, the implementation of minimal-sensor based system was suggested for detecting the fall. The accuracy of the suggested system was calculated 87.56%.

B. -H. Wang, et.al (2020) investigated a new visual-based method for detecting fall in which a Dual-Channel Feature Integration (DCFI) was implemented [31]. This method was emphasized on splitting the fall event into falling-state and fallen-state to define the fall events from dynamic and static viewpoints. First of all, You Only Look Once (YOLO) and the OpenPose models were put forward to pre-process the data so that key points and the position information regarding a human body were acquired. After that, a dual-channel sliding window framework was presented for extracting the dynamic and static features of the human body. In addition, MLP (Multilayer Perceptron) and Random Forest (RF) algorithms were adopted for classifying feature data of both attributes at an individual level. In the end, this method was employed to detect fall. The experimental results revealed that the investigated method yielded an accuracy up to 97.33% on UR dataset and 96.91% on Le2i dataset.

B. -S. Lin, et.al (2022) constructed a neuromorphic computing hardware-based system to reorganize and relocate the neural network (NN) model of the back-end computer into the edge computing (EC) platform [32]. This platform was considered for implementing the considered model with 8-bit precision to transform the object images into human motion attributes. The data was classified using a support vector machine (SVM) method. The experiments exhibited that the constructed system offered 96% accuracy for detecting fall at 11.5 frames per second (FPS) speed and consumed 0.3W of energy. Moreover, this system was capable of monitoring the fall events of older persons in real time, and helped to protect the privacy of the user.

Q. Han, et.al (2020) devised a two-stream technique to detect fall via the MobileVGG algorithm. The motion features of human body were considered in the initial stream for detecting falls and the second one employed an enhanced lightweight VGG algorithm called MobileVGG [33]. This algorithm was built on the basis of integrating point, depth and point convolution. The major objective was to generate a residual association among layers for dealing with the gradient disappearance and the obstructing gradient reflux in the deep model. The experimental results indicated the supremacy of the devised technique over the traditional techniques to detect falls from similar daily activities: lying etc. and mitigate the memory usage. Moreover, the adaptability of this technique was proved for mobile devices.

W. -J. Chang, et.al (2021) introduced an artificial intelligence (AI) technique for detecting falls in which an edge computing (EC) namely the pose estimation-based fall detection methodology (PEFDM) was deployed on the basis of method of recognizing a human body posture [34]. This technique was useful for alleviating the computation load, and its execution was smooth on mainstream EC systems, and AI computing potentials were deployed in this technique. The EC system was utilized for removing the issues related to privacy and upload bandwidth occurred due to image outflow. Based on experiments on real human, the introduced technique was effective for detecting falls for elderly individuals at an accuracy of 98.1%.

3. RESEARCH METHODOLOGY

This research is based on the fall detection and proposed a model which is the combination of CNN and LSTM. The fall is detected in various phases, including dataset collection, pre-processing, and classification. A hybrid deep learning model is designed in this work which is the combination of CNN and Bidirectional LSTM. LSTM network makes the implementation of a micro-gate control by combining short-term and long-term, and provides a solution of gradients disappearing to some level. It consists of three explicit frameworks: a forget gate, an input gate and an output gate. LSTM, by means of a structure termed as gate, is useful to remove or add information to the cell state. In the first step, the information, to be discarded from the cell state, is determined. This issue is solved through a layer known as the forget gate. The forget gate, after reading the hidden state of the earlier moment h_{t-1} and the current input data x_t, leads to generate the output as a vector between 0 and 1. The value between zero and 1 in this vector represents the scale of information which is preserved or rejected in the cell state c_t. The value 0 denotes that all information is rejected while 1 indicates the retention of all information.

$$z_f = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right) \quad (1)$$

The second step aims to determine the scale of new information added to the cell state. This is a two-step process. This step initially decides the information for updating it through an input gate operation. For this, h_{t-1} and x_t are used. Then, h_{t-1} and x_t are used for obtaining a new candidate cell state z via a tanh layer.

$$z_i = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (2)$$

$$z = tanh(W.[h_{t-1}, x_t] + b) \quad (3)$$

Next, the cell state is updated as follow:

$$c_t = z_f \odot c_{t-1} + z_i \odot z. \quad (4)$$

The final step is to determine the output value. Once the cell state is updated, the part of the cell state which is obtained as the output depending upon the input h_{t-1} and x_t, is decided [20]. For this purpose, the input is passed through a sigmoid layer known as output gate so that the judgment conditions can be obtained. After this, there is a need to transmit the cell state via the tanh layer to get a vector between -1 and 1. Finally, the output is achieved by multiplying this vector with the analysis conditions attained from the output gate.

$$z_o = \sigma(W_O.[h_{t-1}, x_t] + b_0) \quad (5)$$

$$h_t = z_o \odot tanh(c_t) \quad (6)$$

In the above expression, z_i is the forget gate, z_f is the input gate, and z_o is the output gate. Apart from this, z denotes the input via a tanh layer. This is also termed as a candidate cell state. At last, $\odot$ represents a multiplication function of the related components of the matrix. CNN is a kind of Deep Learning algorithm, utilizes to detect the data. Similar to ML

(Machine Learning) algorithms, this algorithm is capable of learning the metrics for which a training set is employed so that least empirical and structural risk is obtained. Hence, the LF (loss function) is alleviated. Diverse operations support distinct LFs to represent the error occurred in predictive values and value having a label 'correct'. DL is useful to stack the layers of metrics and establish an association amid them and AF (activation function). After that, the network becomes adaptable for the nonlinear functions which have complexity. Unlike the traditional NNs (neural networks), there are 3 layers for building the Convolutional Neural Network which are Conv (convolutional), Pooling and FC (fully connected) layers. The initial layer is employed for mapping an input of multilayer into an output. Every image layer is considered as a channel. The mapping task is accomplished using a kernel for every channel. A convolution is exploited with the kernel with the purpose of separating every layer into small units (5×5) and generating a number from every unit. The convolution of every unit is expressed as:

$$E_{p,q} = \sum_{i,j} A_{ij} \cdot W_{ij} \tag{7}$$

In this, a square matrix based on (p, q) is represented with A, the correspondence components are defined through A_{ij} andW_{ij} and the output is obtained in the form of $E_{p,q}$. The sigmoid function (SF) or rectified linear unit (ReLU) function plays a role of an activation function to make the mapping non-linear. The given equation defines the ReLU function:

$$f(x) = \begin{cases} x, & x >= 0 \\ 0, & x < 0 \end{cases} \tag{8}$$

This layer aims to abstract the classic attributes from the input. The lower layers are executed for retrieving the horizontal or vertical edges and the upper layers emphasize on integrating these attributes.

```
=================================================================
 conv1d_26 (Conv1D)          (None, 24, 64)            384

 activation_30 (Activation)  (None, 24, 64)            0

 dropout_13 (Dropout)        (None, 24, 64)            0

 max_pooling1d_10 (MaxPoolin  (None, 6, 64)            0
 g1D)

 conv1d_27 (Conv1D)          (None, 6, 128)            41088

 activation_31 (Activation)  (None, 6, 128)            0

 dropout_14 (Dropout)        (None, 6, 128)            0

 max_pooling1d_11 (MaxPoolin  (None, 1, 128)           0
 g1D)

 conv1d_28 (Conv1D)          (None, 1, 256)            164096

 activation_32 (Activation)  (None, 1, 256)            0

 dropout_15 (Dropout)        (None, 1, 256)            0

 bidirectional (Bidirectiona  (None, 128)              164352
 l)

 dense_8 (Dense)             (None, 2)                 258

 activation_33 (Activation)  (None, 2)                 0

=================================================================
Total params: 370,178
```

Figure 3. Model Description

4. RESULT & DISCUSSION

The videos of 30 participants engaging in everyday tasks are considered in this study while wearing a smartphone strapped on their waist with inertial sensors for generating an HAR data set. The major goal is to assign activities to one of 6 done activities. Thirty volunteers are considered in the simulation. Every participant makes the deployment of a smartphone (Samsung Galaxy S II) while doing 6 activities (walking, climbing stairs, walking down stairs, sitting, standing, and lying). Its integrated accelerometer and gyroscope is utilized to record 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments are executed on videotapes in order to assign labels to the data in manual way. The resulting data set is divided into 2 sets randomly, with 30% of the volunteers selected for generating the testing data and 70% for training data.

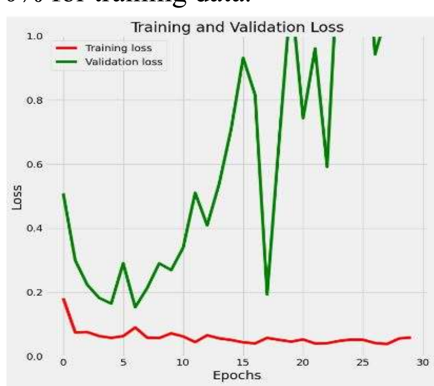

Figure 4. Training and Validation Loss

As shown in figure 4, the training validation loss the proposed model is shown. It is analysed that training accuracy is achieved above 95 percent.

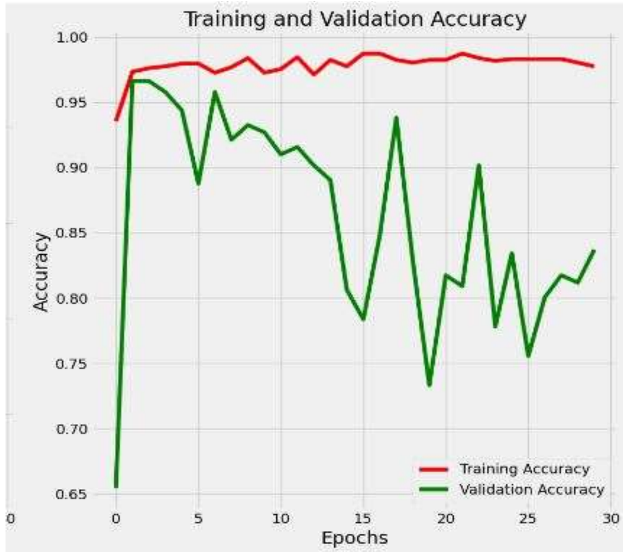

Figure 5. Training and Validation Accuracy

As shown in figure 5, the training and validation accuracy of the proposed model is shown. It is analysed that training accuracy is above 95 Percent.

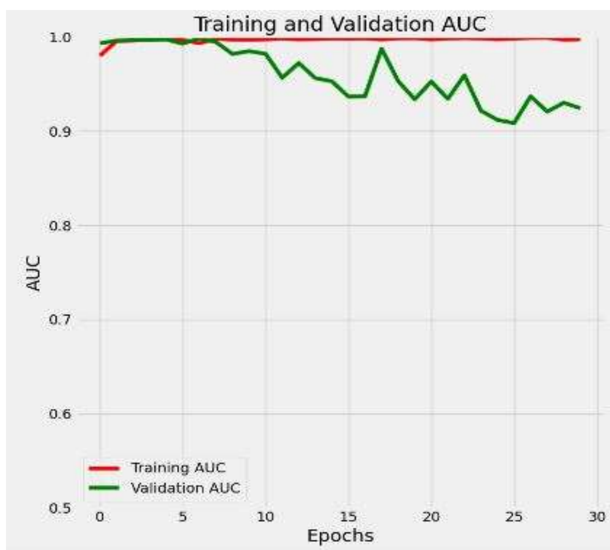

Figure 6. Training and Validation AUC

As shown in figure 6, the training and validation AUC is illustrated. The training and validation AUC is approx. 99 percent.

Table 1. Performance Analysis

Model	Accuracy	Precision	Recall
SVM	79.56 Percent	79 Percent	79 Percent
KNN	77.12 Percent	77 Percent	77 Percent
Proposed	87.16 Percent	87 Percent	87 Percent

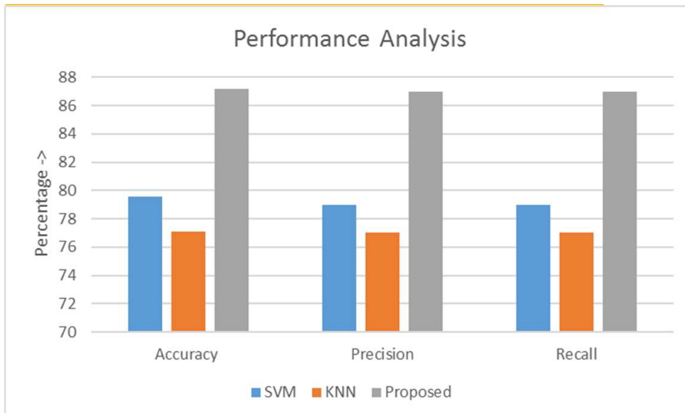

Figure 7. Performance Analysis

As shown in figure 7, the performance of proposed model is compared with SVM and KNN. The proposed model achieves an accuracy of upto 87.16 percent as compared to SVM and KNN.

5. CONCLUSION

Many applications are available for human fall detection, such as intelligent video surveillance, home environmental monitoring, and to store and retrieve the video, etc. In this research work, a dataset is obtained through kaggle which is an authorized repository to collect a dataset. Missing values from the gathered data are removed when the data is pre-processed. The obtained dataset is then further processed for extracting the attributes. The deep learning model is proposed for the human fall detection. The proposed model is a combination of CNN and LSTM. Two models: CNN and LSTM are also implemented for detecting the human fall. The proposed hybrid model has performed well as compared to existing models concerning accuracy, precision and recall. The proposed model achieves accuracy, precision and recall of upto 87.5 percent for detecting human fall.

REFERENCES

[1] H. Zhang, J. Liu, and N. Kato, "Threshold Tuning-Based Wearable Sensor Fault Detection for Reliable Medical Monitoring Using Bayesian Network Model," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1886–1896, 2018, doi: 10.1109/JSYST.2016.2600582.

[2] D. Ajerla, S. Mahfuz, and F. Zulkernine, "A real-time patient monitoring framework for fall detection," *Wirel. Commun. Mob. Comput.*, vol. 2019, 2019, doi:

10.1155/2019/9507938.

[3] A. El Attaoui, S. Largo, S. Kaissari, A. Benba, A. Jilbab, and A. Bourouhou, “Machine learning-based edge-computing on a multi-level architecture of WSN and IoT for real-time fall detection,” *IET Wirel. Sens. Syst.*, vol. 10, no. 6, pp. 320–332, 2020, doi: 10.1049/iet-wss.2020.0091.

[4] Y. K. Kang, H. Y. Kang, and J. B. Kim, “A Study of Fall Detection System Using Context Cognition Method,” *Proc. - 2021 21st ACIS Int. Semi-Virtual Winter Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD-Winter 2021*, pp. 79–83, 2021, doi: 10.1109/SNPDWinter52325.2021.00024.

[5] X. Wang, J. Ellul, and G. Azzopardi, “Elderly Fall Detection Systems: A Literature Survey,” *Front. Robot. AI*, vol. 7, no. June, 2020, doi: 10.3389/frobt.2020.00071.

[6] P. Vallabh and R. Malekian, “Fall detection monitoring systems: a comprehensive review,” *J. Ambient Intell. Humaniz. Comput.*, vol. 9, no. 6, pp. 1809–1833, 2018, doi: 10.1007/s12652-017-0592-3.

[7] T. De Quadros, A. E. Lazzaretti, and F. K. Schneider, “A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device,” *IEEE Sens. J.*, vol. 18, no. 12, pp. 5082–5089, 2018, doi: 10.1109/JSEN.2018.2829815.

[8] M. Islam *et al.*, “Deep Learning Based Systems Developed for Fall Detection: A Review,” *IEEE Access*, vol. 8, no. September, pp. 166117–166137, 2020, doi: 10.1109/ACCESS.2020.3021943.

[9] O. Aziz, M. Musngi, E. J. Park, G. Mori, and S. N. Robinovitch, “A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials,” *Med. Biol. Eng. Comput.*, vol. 55, no. 1, pp. 45–55, 2017, doi: 10.1007/s11517-016-1504-y.

[10] M. Saleh, M. Abbas, J. Prud’Homm, D. Somme, and R. Le Bouquin Jeannes, “A Reliable Fall Detection System Based on Analyzing the Physical Activities of Older Adults Living in Long-Term Care Facilities,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 2587–2594, 2021, doi: 10.1109/TNSRE.2021.3133616.

[11] D. Yu and J. Li, “Recent progresses in deep learning based acoustic models,” *IEEE/CAA J. Autom. Sin.*, vol. 4, no. 3, pp. 396–409, 2017, doi: 10.1109/JAS.2017.7510508.

[12] J. S. Lee and H. H. Tseng, “Development of an Enhanced Threshold-Based Fall Detection System Using Smartphones with Built-In Accelerometers,” *IEEE Sens. J.*, vol. 19, no. 18, pp. 8293–8302, 2019, doi: 10.1109/JSEN.2019.2918690.

[13] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, “A smartphone-based fall detection system,” *Pervasive Mob. Comput.*, vol. 8, no. 6, pp. 883–899, 2012, doi: 10.1016/j.pmcj.2012.08.003.

[14] T. Elfaramawy, C. L. Fall, S. Arab, M. Morissette, F. Lellouche, and B. Gosselin, “A Wireless Respiratory Monitoring System Using a Wearable Patch Sensor Network,” *IEEE Sens. J.*, vol. 19, no. 2, pp. 650–657, 2019, doi: 10.1109/JSEN.2018.2877617.

[15] E. Valero, A. Sivanathan, F. Bosché, and M. Abdel-Wahab, “Analysis of construction trade worker body motions using a wearable and wireless motion sensor network,” *Autom. Constr.*, vol. 83, no. August, pp. 48–55, 2017, doi:

10.1016/j.autcon.2017.08.001.

[16] M. Tang and D. Ou, "Fall Detection System for Monitoring an Elderly Person Based on Six-Axis Gyroscopes," vol. 127, no. Eame, pp. 244–247, 2018, doi: 10.2991/eame-18.2018.51.

[17] S. Iloga, A. Bordat, J. Le Kernec, and O. Romain, "Human Activity Recognition Based on Acceleration Data from Smartphones Using HMMs," *IEEE Access*, vol. 9, pp. 139336–139351, 2021, doi: 10.1109/ACCESS.2021.3117336.

[18] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The MobiAct dataset: Recognition of activities of daily living using smartphones," *ICT4AWE 2016 - 2nd Int. Conf. Inf. Commun. Technol. Ageing Well e-Health, Proc.*, no. Ict4awe, pp. 143–151, 2016, doi: 10.5220/0005792401430151.

[19] Y. Z. Hsieh and Y. L. Jeng, "Development of Home Intelligent Fall Detection IoT System Based on Feedback Optical Flow Convolutional Neural Network," *IEEE Access*, vol. 6, pp. 6048–6057, 2017, doi: 10.1109/ACCESS.2017.2771389.

[20] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil, "A Study on Human Fall Detection Systems: Daily Activity Classification and Sensing Techniques," *Int. J. Integr. Eng.*, vol. 8, no. 1, pp. 35–43, 2016, [Online]. Available: http://penerbit.uthm.edu.my/ojs/index.php/ijie/article/view/1332/911

[21] Y. Wu, Y. Xiao, and H. Ge, "Fall detection monitoring system based on MEMS sensor," *J. Phys. Conf. Ser.*, vol. 1650, no. 2, 2020, doi: 10.1088/1742-6596/1650/2/022037.

[22] D. Pan, D. Pan, and D. Qu, "Heterogeneous Sensor Data Fusion for Human Falling Detection," *IEEE Access*, vol. 9, pp. 17610–17619, 2021, doi: 10.1109/ACCESS.2021.3051899.

[23] C. Bicchielli, "Multi-Sensor Fall Detection for Smartphones," *Biomed. J. Sci. Tech. Res.*, vol. 34, no. 4, pp. 26978–26992, 2021, doi: 10.26717/bjstr.2021.34.005588.

[24] Y. Harari, N. Shawen, C. K. Mummidisetty, M. V. Albert, K. P. Kording, and A. Jayaraman, "A smartphone-based online system for fall detection with alert notifications and contextual information of real-life falls," *J. Neuroeng. Rehabil.*, vol. 18, no. 1, pp. 1–13, 2021, doi: 10.1186/s12984-021-00918-z.

[25] X. Lv, Z. Gao, C. Yuan, M. Li, and C. Chen, "Hybrid Real-Time Fall Detection System Based on Deep Learning and Multi-sensor Fusion," *Proc. - 2020 6th Int. Conf. Big Data Inf. Anal. BigDIA 2020*, pp. 386–391, 2020, doi: 10.1109/BigDIA51454.2020.00069.

[26] T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. Diaz, I. V. Pustokhina, and D. A. Pustokhin, "Internet of Things and Deep Learning Enabled Elderly Fall Detection Model for Smart Homecare," *IEEE Access*, vol. 9, pp. 113879–113888, 2021, doi: 10.1109/ACCESS.2021.3094243.

[27] Z. Mohammad, A. R. Anwary, M. F. Mridha, M. S. H. Shovon, and M. Vassallo, "An Enhanced Ensemble Deep Neural Network Approach for Elderly Fall Detection System Based on Wearable Sensors," *Sensors*, vol. 23, no. 10, pp. 1–17, 2023, doi: 10.3390/s23104774.

[28] A. Choi *et al.*, "Deep Learning-Based Near-Fall Detection Algorithm for Fall Risk Monitoring System Using a Single Inertial Measurement Unit," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2385–2394, 2022, doi: 10.1109/TNSRE.2022.3199068.

[29] D. M. Ding, Y. G. Wang, W. Zhang, and Q. Chen, "Fall Detection System on Smart Walker Based on Multisensor Data Fusion and SPRT Method," *IEEE Access*, vol. 10, no. August, pp. 80932–80948, 2022, doi: 10.1109/ACCESS.2022.3195674.

[30] H. Ponce, L. Martínez-Villaseñor, and J. Nuñez-Martínez, "Sensor location analysis and minimal deployment for fall detection system," *IEEE Access*, vol. 8, pp. 166678–166691, 2020, doi: 10.1109/ACCESS.2020.3022971.

[31] B. H. Wang, J. Yu, K. Wang, X. Y. Bao, and K. M. Mao, "Fall Detection Based on Dual-Channel Feature Integration," *IEEE Access*, vol. 8, pp. 103443–103453, 2020, doi: 10.1109/ACCESS.2020.2999503.

[32] B. S. Lin *et al.*, "Fall Detection System with Artificial Intelligence-Based Edge Computing," *IEEE Access*, vol. 10, pp. 4328–4339, 2022, doi: 10.1109/ACCESS.2021.3140164.

[33] Q. Han *et al.*, "A two-stream approach to fall detection with mobilevgg," *IEEE Access*, vol. 8, pp. 17556–17566, 2020, doi: 10.1109/ACCESS.2019.2962778.

[34] W. J. Chang, C. H. Hsu, and L. B. Chen, "A Pose Estimation-Based Fall Detection Methodology Using Artificial Intelligence Edge Computing," *IEEE Access*, vol. 9, pp. 129965–129976, 2021, doi: 10.1109/ACCESS.2021.3113824.