Detecting Falls with Wearable Sensors Using Machine Learning Techniques

Shailendra Bhandari, Negar Elmisadr, Bereket Zerabruk Tekeste and Raju Shrestha

May 13, 2023

Detecting Falls with Wearable Sensors using Machine Learning Techniques

1st Shailendra Bhandari
Department of Computer Science
OsloMet – Oslo Metropolitan University
Oslo, Norway
shailendra.vandari@gmail.com/ orcid.org/0000-0002-7860-4854

2ndNegar Elmisadr
Department of Computer Science
OsloMet– Oslo Metropolitan University
Oslo, Norway
s366271@oslomet.no

3rdBereket Zerabruk Tekeste
Department of Computer Science
OsloMet– Oslo Metropolitan University
Oslo, Norway
s331401@oslomet.no

4thRaju Shrestha
Department of Computer Science
OsloMet– Oslo Metropolitan University
Oslo, Norway
raju.shrestha@oslomet.no

***Abstract*—This study presents an innovative approach to the problem of fall detection, leveraging wearable sensor technology. Initially, we delineate the definition of falls, followed by an examination of their classification methods and categories. To address this issue, we propose a typology that employs both Long Short-Term Memory (LSTM) and a hybrid Convolutional Neural Network (CNN1D + LSTM). These models are trained to detect falls, utilizing data sourced from accelerometers, gyroscopes, and magnetometers. Our proposed network models are trained and rigorously evaluated using a comprehensive wearable sensor dataset. They are also benchmarked against a range of different classifiers for comparative purposes. The LSTM model demonstrated an impressive accuracy of 98.04%, while the hybrid CNN1D + LSTM model achieved an exceptional accuracy of 99.68%. To further validate our approach, we compared the performance of our models against other deep neural network architectures that have previously been proposed and implemented. Our models demonstrated competitive, if not superior, performance, endorsing their potential for effective real-world application in fall detection.**

***Index Terms*—Deep Learning, Neural Networks, Fall Detection**

I. Introduction

Over the past two decades, automatic fall detection (FD) devices have evolved as crucial assistive technologies. The primary function of these systems is to promptly detect significant falls and alert medical professionals or caregivers, thereby mitigating potential risks. Moreover, these solutions can also alleviate the psychological stress associated with caregiving for the elderly. It is well-established that individuals aged 60 and above are susceptible to adverse outcomes following falls, necessitating effective real-time detection and analysis.

Numerous studies have explored automatic fall detection using various technologies, including sensors, video monitoring, and wearable devices. Decision-making based on data from multiple sources often proves more effective than relying on a single source. Consequently, multimodal fall detection, leveraging data from diverse sources for accurate fall identification, has gained traction in research.

Fall detection is of paramount importance in fields such as preventive medicine, wellness management, and assisted living, particularly for the elderly population. Consequently, an array of fall detection systems has emerged, either reported in the scientific literature or commercially available. Most of these systems predominantly depend on accelerometers and gyroscopes attached to an individual's body. These systems utilize either discrete sensors incorporated into a device specifically designed for this purpose or sensors integrated with smartphones. The latter offers established and universally accessible communication facilities, such as the ability to contact emergency services. Despite these advancements, automatic fall detection continues to face significant challenges, with the most formidable being the accurate determination of the type of fall. As this field continues to progress, addressing these challenges will be essential for enhancing the effectiveness and reliability of fall detection systems.

Various sensor technologies [1], such as inertial sensors, depth cameras, microphones, pressure sensors, and thermal sensors, have been used in fall detection systems due to improvements in information transmission technologies and body sensor networks. Accelerometers, which capture body motions and are sensitive to posture changes, are the most prevalent sensors for fall detection. Accelerometer-based fall detection systems have several advantages, including compactness, low cost, effectiveness, unobtrusiveness, and high mobility.

Wearable sensors must be implanted in the body to provide long-term fall detection services and function for as long as possible. This need leads to issues in the design and development of systems, including reliability, security, usability, and sustainability. The amount of batteries, for example, has an impact on the size and comfortability of sensors. Furthermore, repeatedly recharging or replacing batteries may reduce the fall detection system's utility and user acceptance. As a result, numerous energy-efficient fall detection systems have been developed for long-term fall detection services that aim at minimizing power consumption and extending battery life as long as feasible to enable long-term fall detection services. This need creates system design and development issues, such as reliability, security, usability, and long-term viability.

Even though numerous fall detectors have been used and some are on the market, none has been labeled as a superior method. According to estimates, nearly half of the elderly who fall do not disclose it to their healthcare provider. This fact encourages the usage of non-wearable gadgets with remote monitoring capabilities. A computer system that can identify and classify falls automatically and effectively would help monitor the older population and speed up the help process, minimizing the risk of long-term injury and mortality. One of the most typical issues with such systems is a high number of false positives in their recognition method, leading to a surge in surveillance system calls. According to current multi-disciplinary research, falls among the elderly are a significant global public health concern. Several wearable motion sensor-based fall detection systems have been developed, but these systems fail to adequately assess the exact nature of human falls. Fall detection devices are less expensive than a person's daily observation. A form of fall device reported to be in use is a wearable sensor device that consists of a magnetometer, gyroscope, and accelerometer tri-axial device. This device detects a fall with the utmost accuracy and simplicity.

In the aged, falls are majorly responsible for severe injuries and death. According to the World Health Organization [2], over 30% of the elderly, aged 64 and more, fall at least once a year. A previous study showed that nearly half of the elderly population died after laying on the floor for more than an hour within six months of a fall. Also, around 420,000 falls result in death each year. As a result of this statistic, falls are the second most significant cause of unintentional injury mortality.

II. Related Works

Various papers give an account of the development of fall detection from different aspects. We chose the most highly cited review papers from 2014 to 2020. There are two approaches to fall detection using wearable sensors—threshold-based systems and machine learning-based systems. While threshold-based systems have been popular because of their low computational overhead, they could be prone to more false positives and false negatives, given that the thresholds themselves may be affected by various factors. As a result, machine learning algorithms for fall detection have been a much-researched area. There has been extensive research into the efficiencies of various machine-learning techniques for fall detection.

de Quadros et al [3] compare threshold-based mechanisms and machine learning-based mechanisms for fall detection applied on data generated by accelerometer, gyroscope, and magnetometer. The paper concludes that the machine learning-based mechanism yielded much better results than the threshold-based solutions. Machine learning-based techniques differ from each other in multiple factors—the feature set used, sensors employed, placement of sensors, algorithms applied, the dataset used, performance parameters monitored, and so on. In [4], the dataset was generated from an accelerometer and gyroscope at the waist level. Feature extraction was performed using the windowing technique, feature selection using the rank-based system, and classification using Naïve-Bayes, LSM, ANN, SVM, and kNN algorithms. kNN, ANN, and SVM had the best performance results compared to LSM and Naïve-Bayes. Results show an accuracy of 87.5%, a sensitivity of 90.70%, and a specificity of 83.78%, for kNN.

Jefiza et al. [5] use a backpropagation neural network (BPNN) for fall detection, with data collected from a three-axis accelerometer and gyroscope, and reported an accuracy of 98.182%, a precision of 98.33%, the sensitivity of 95.161%, and specificity of 99.367%. Hossain et al. [6] also attempt to distinguish falls from ADLs and compare SVM, kNN, and complex tree algorithms applied to data generated by accelerometers. The paper compared the performance of these algorithms concerning the accuracy, precision, and recall of ADLs and four types of falls (forward, backward, right, and left). It was observed that the accuracy and precision of SVM were the highest, while complex trees performed better in terms of recall analysis.

One of the observed drawbacks of wearable sensors is that the placement of the sensors impacts the accuracy of fall classification and detection. Yu et al. [7] attempt to reduce errors caused by incorrect sensor positions and detail an HMM-based fall detection system for the same. Sensor orientation calibrations are applied on HMM classifiers to resolve issues arising out of misplaced sensor (3-axis accelerometer) locations and misaligned sensor orientations. This paper reports a sensitivity of 99.2% on an experimental dataset and 100% for a real-world fall dataset.

Chelli et al. [8] compares the performance of 4 algorithms—ANN, kNN, quadratic SVM, and ensemble bagged tree—in two steps. First, only acceleration and angular velocity data are used. Then, new features that improve the performance of the classifier are extracted from the power spectral density of the acceleration. The accuracy of the algorithms is observed to have increased after applying feature extraction techniques. The objective of Wang et al. [9] was to test the impact of optimal feature selection on fall detection accuracy. The features of accelerations in different parts of the body are collected through wearable devices. The Bayesian framework was applied to select the optimal features from the data generated by the wearable devices. The weight of each feature was calculated, after which training was done based on the optimal feature set. It was observed that improved classification led to better accuracy, sensitivity, and specificity.

Genoud et al. [10] propose a system for soft fall detection using ML in wearable devices. The feature sets used were linear acceleration and gyroscope readings, and the algorithms compared were decision tree, decision tree ensemble, kNN, and multilayer perceptrons (MLP). The experiments showed that the decision tree ensemble outperformed the results obtained by the other algorithms. Kao et al. [11] use an ensemble of spectrum analysis with GA-SVM, SVM, and C4.5 classifiers. The sensor readings were from 3-axis accelerometers. The best results were given by GA-SVM, with an accuracy of 94.1%, a sensitivity of 94.6%, and a specificity of 93.6%.

Musci et al. [12] describe an RNN model with LSTM blocks on data generated by 3D accelerometers for fall detection. The paper observes that though it is difficult to distinguish high dynamic activities from falls, the approach described achieves a better overall classification. Fakhrulddin et al. [13] apply CNN to streaming time series accelerometer data collected from body sensor networks (BSN) for fall and non-fall situations. Yves M. Galvão et al. [14] proposed a multimodal approach with a convolution neural network and LSTM trained to detect falls based on RGB images and information from accelerometers and provide an extensive comparison with state-of-the-art models, a multimodal solution presents an improvement in the accuracy.

III. A Deep Learning Model for Fall Detection

A. *Convulational Neural Network*

Convolutional Neural Networks (CNNs) are a type of deep neural network that apply convolution operations to learn kernels for extracting features from input data. These kernels can be defined using different topological models, such as 1D kernels primarily used for temporal processing within a defined window and 2D kernels predominantly used for spatial relation learning and image processing [15]–[17]. CNNs have been instrumental in addressing a variety of challenges, including decoding facial recognition, analyzing documents, and understanding climate grey areas. A distinguishing characteristic of CNNs, in comparison to conventional neural networks, is their superior performance with image, speech, or audio signal inputs.

CNNs are typically composed of three types of layers: convolutional, pooling, and fully connected. The convolutional layer, which is the first and most crucial component, performs most of the computation. This layer requires input data, a filter, and a feature map, among other components. The input, often a 3D matrix of pixels from a color image, is subjected to a convolution operation using a 2D weighted array known as a feature detector, kernel, or filter. This filter systematically sweeps across the image, performing dot products of the input pixels and the filter values. The outputs of these operations, known as feature maps or convoluted features, are then passed to subsequent layers.

Following the convolutional layers, the pooling or down-sampling layers serve to reduce the dimensionality of the data, thereby enhancing computational efficiency and mitigating overfitting. This layer applies a filter across the input and populates the output array based on a clustering method. Two common methods used are max pooling, where the highest value pixel in the receptive field is selected, and average pooling, where the mean value of the receptive field is calculated.

The final layer is the fully-connected layer, where each node in the output layer is directly connected to a node in the preceding layer, contrasting with the partially connected layers where the pixel values of the input image are not directly connected to the external layer.

The sequential combination of these layers allows CNNs to process data hierarchically, focusing on fundamental features like colors and edges in the initial layers and identifying increasingly complex attributes as data progresses through the network. This hierarchical processing enables CNNs to accurately identify the target object, highlighting the sophisticated design of CNNs in facilitating nuanced image recognition tasks.

B. *Long-Short Term Memory*

The long short-term memory (LSTM) architecture is a deep learning architecture that involves a recurrent neural network (RNN) published [18] in 1997 by Sepp. Hochreiter and Jürgen Schmidhuber. Unlike regular feed-forward neural networks, LSTM has feedback connections. Therefore, it is well suited to learn from experience to classify, process, and predict time series when they are very long time lags of unknown size between important events. A memory cell in LSTM is composed of four main elements, an input gate, a neuron with self recurrent connection, a forget gate, and an output gate. An input gate can allow incoming signals to alter the state of the memory cell or block it. The reconnect connection bars any outside inference where the state of the memory cell can remain constant from one step to another. The forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state as needed. And finally, the output gate can allow the memory cell's state to affect other neurons or prevent it. In this project work, we are employing single-layer LSTM as well as multi-layer LSTM RNN. The network architecture for the LSTM model we used in this dataset is shown in Figure [1].

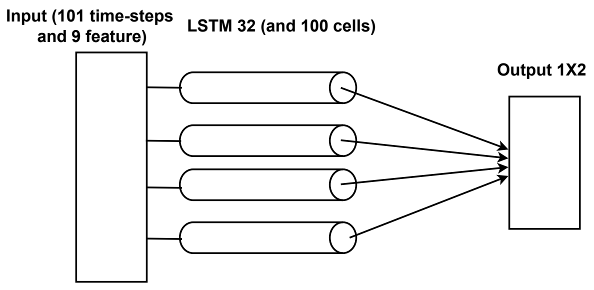


Fig. 1. Architecture used for the Dataset Accelerometer— LSTM.

IV. Experimental Setup

A. *Dataset*

The data set for this project is taken from the reference article [19] and is available publicly in the link [1]. With

[1]https://drive.google.com/open?id=1gqS1fkTvtuAaKj_0cn9n04ng1qDAoZ2t

Erciyes University Ethics Committee approval, ten males (24 ± 3 years old, 67.5 ± 13.5 kg, 172 ± 12 cm) and seven females (21.5 ± 2.5 years old, 58.5 ± 11.5 kg, 169.5 ± 12.5 cm) healthy volunteers participated in the study with informed written consent. A wireless sensor unit was fitted to the subject's waist and right thigh, among other body parts. The sensor unit comprises three tri-axial devices: an accelerometer, a gyroscope, and a magnetometer/compass. Raw motion data were recorded along three perpendicular axes (x, y, z) from the unit with a sampling frequency of 25 Hz yielding $Acc_X, Acc_Y, Acc_Z(m/s^2)$, $Gyr_X, Gyr_Y, Gyr_Z(/s)$ and $Mag_X, Mag_Y, Mag_Z(Gauss)$. The data set consists of 57.96% falls and 42.04% of activities of daily life. In addition, there are altogether 1570 records, 910 falls, and 660 daily life activities.

B. *Pre-processing*

Usually, the studies on fall detection mostly use simple thresholding of the sensory outputs like acceleration and rotational rates because of its simplicity and low processing time. However, in this dataset, additional features of the recorded signals are considered. The total acceleration of the waist accelerometer is given by

$$A_T = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

where A_x, A_y, A_z are the accelerations along the x, y, and z axes respectively. Initially, the time index corresponding to the peak AT value of the waist accelerometer in each record was identified. Then, the two-second intervals $(25Hz \times 2s = 50$ samples) before and after this point was taken, corresponding to a time window of 101 samples $(50 + AT \quad index + 50)$ and ignore the rest of the record. Data from the remaining axes of each sensor unit are also reduced in the same way, considering the time index obtained from the waist sensor as a reference, resulting in six 101 × 9 arrays of data. Each column of data is represented by an $N \times 1$ vector $s = [s1, s2, \ldots, sN]T$, where $N = 101$. Extracted features consist of the minimum, maximum, and mean values, as well as variance, skewness, kurtosis, the first 11 values of the auto-correlation sequence, and the first five peaks of the discrete Fourier transform (DFT) of the signal with the corresponding frequencies.

The total acceleration of the five falls plotted over a four-second time interval around their peak at time 0 is shown in Figure [2]. The individual fall differs from one another, as shown in the figure. Similarly, Figure [3] shows the mean total acceleration of all 910 falls.

The total five activities of daily life plotted over a four-second time interval around their peak at time 0 is shown in Figure [4]. The individual activities of daily life differ from one another as shown in the figure. Similarly, Figure [5] shows the mean total acceleration of all 660 activities of daily life.

C. *Evaluation protocol*

From the raw data, nine measure signals Acc_X, Acc_Y, Acc_Z, Gyr_X, Gyr_Y, Gyr_Z, Mag_X, Mag_Y, Mag_Z) of the $FallDataSet$ and 17 features, 153 (17×9) feature vector of dimensionality for each test are extracted. Feature extraction is essential for implementation with standard machine learning classifiers. This will be briefly explained in the result section. Extraction is unnecessary for deep neural network features as we are feeding the raw data. We evaluate all the models by training 80% of the data point from the datasets and testing the remaining data points. We investigated three different topology architectures for processing the raw sensor data to optimize our performance. The proposed network model architectures are based on LSTM shown in Figure [6,7] and LSTM CNN1D as shown in Figure [8]. Using the CNN and LSTM model, we evaluate the accelerometer, magnetometer, and gyroscope data processing with a one-dimensional filter CNN1D and the LSTM. CNN is an expert in processing spatial relations, while LSTM is useful for the processing of temporal as well as spatial patterns. Since the data we are analyzing is not a multi-class dataset, we are using the sigmoid activation function in the dense layer and binary cross-entropy loss. We use Adam optimizer in all the models. Similarly, we used the ReLu activation function on the input Conv1D layer in Conv1D + LSTM model.

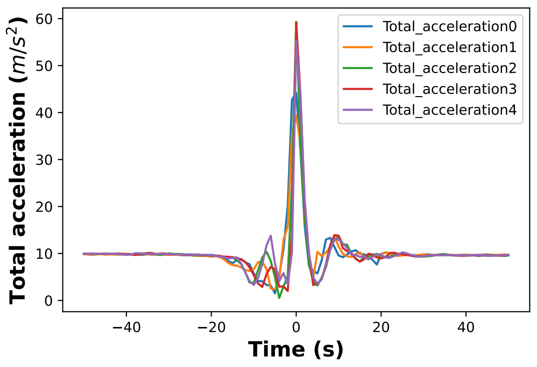


Fig. 2. The total acceleration of five falls plotted over four-second time intervals around their peak at time 0.

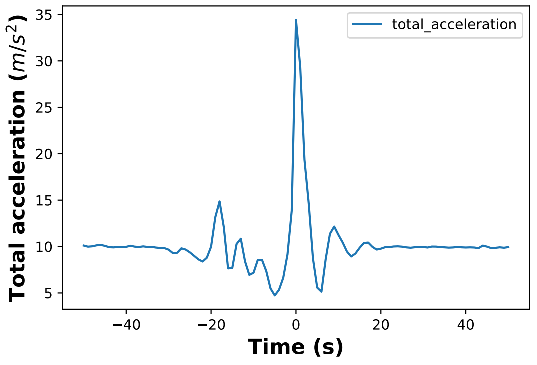


Fig. 3. The accelerations of falls plotted over a time interval.

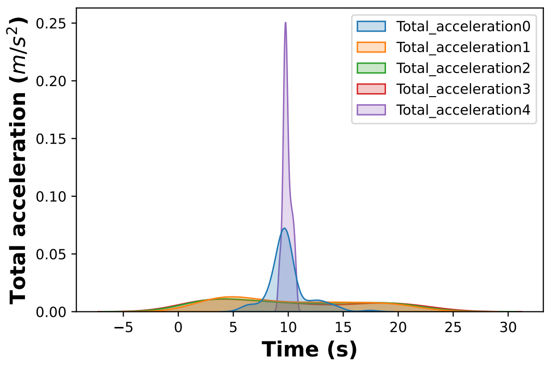


Fig. 4. The total acceleration of five falls plotted over four-second time intervals around their peak at time 0.

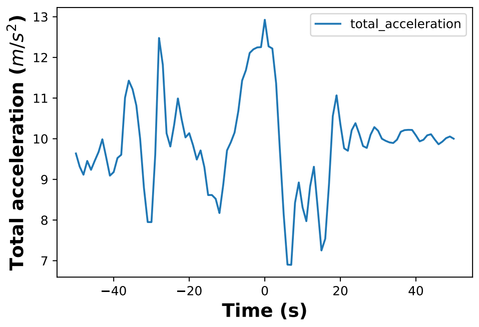


Fig. 5. The accelerations of activities of daily life plotted over a time interval.

V. Results

A. Standard Machine Learning Classifiers

Typical machine learning algorithms, such as support vector machine (SVM), decision tree (DT), random forest (RF), and K- Nearest Neighbours (K-NN), are used to build fall detection models. We extracted the minimum, maximum, mean, median, skewness, kurtosis, and variance features for the three-axis acceleration, velocity, and angular velocity data, trained by the use of the TensorFlow deep learning framework. We propose a multimodal approach by combining Conv1D and LSTM networks with these features. To test the accuracy of the approach in this study, a comparative test experiment with different standard machine learning classification methods was performed. The resulting accuracy scores on the test dataset are reported in Listing [1] and are shown in Figure [9]. The best-performing classifier is the RF, K-NN classifier yielding an accuracy score of 100%.

Listing 1. The standard machine learning classifiers scores

```
         Classifier         Score
0        SVC                0.949045
1        Decision Tree      0.987261
2        Random Forest      1.000000
3        K Neighbors        1.000000
```

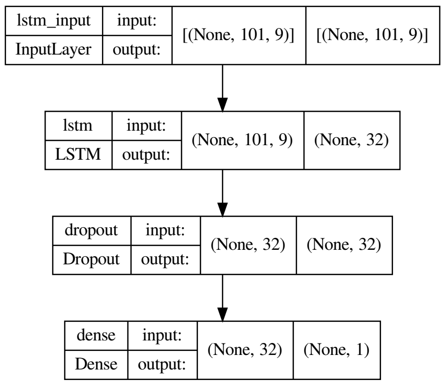

lstm_input | input: | [(None, 101, 9)]
InputLayer | output: | [(None, 101, 9)]
lstm | input: | (None, 101, 9)
LSTM | output: | (None, 32)
dropout | input: | (None, 32)
Dropout | output: | (None, 32)
dense | input: | (None, 32)
Dense | output: | (None, 1)

Fig. 6. The network architecture for single-layer LSTM based model.

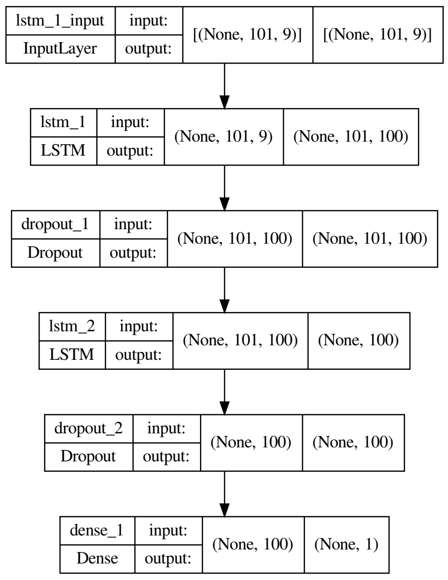

lstm_1_input | input: | [(None, 101, 9)]
InputLayer | output: | [(None, 101, 9)]
lstm_1 | input: | (None, 101, 9)
LSTM | output: | (None, 101, 100)
dropout_1 | input: | (None, 101, 100)
Dropout | output: | (None, 101, 100)
lstm_2 | input: | (None, 101, 100)
LSTM | output: | (None, 100)
dropout_2 | input: | (None, 100)
Dropout | output: | (None, 100)
dense_1 | input: | (None, 100)
Dense | output: | (None, 1)

Fig. 7. The network architecture for multi-layer LSTM based model.

Initially, the accuracy of the K-NN model is 100%; however, in an attempt to reduce the number of features from a total of 153 to 68, the accuracy is still 99.68%. We performed principal component analysis on the training feature dataset to obtain 68 dimensions explaining most of the training data variance. Dimensionality reduction of the full feature space from 153 to 68 features reduces the accuracy score by 0.32%, which is not a drastic reduction, and we should consider a successful simplification. The results from all the evaluated models- K-NN, single and multi-layer LSTM, and Conv1D + LSTM model are summarized in Table [I]. The Conv1D + LSTM model outperforms multi-layer LSTM, followed by the single-layer LSTM in accuracy. Also, the comparison between

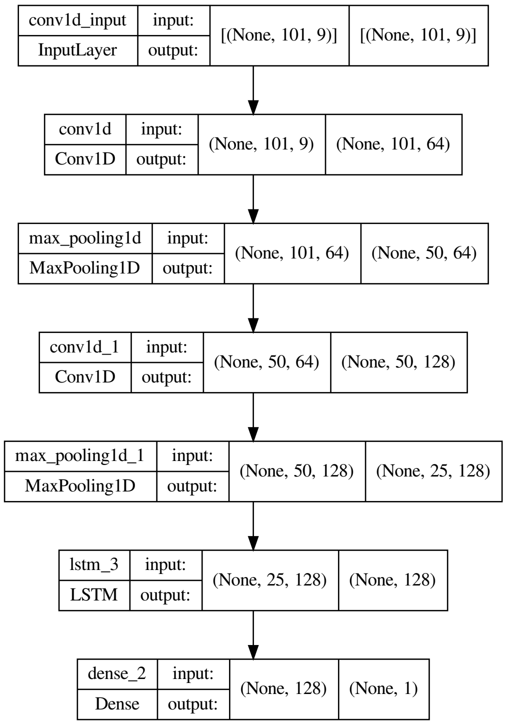


Fig. 8. The network architecture for CNN1D based model.

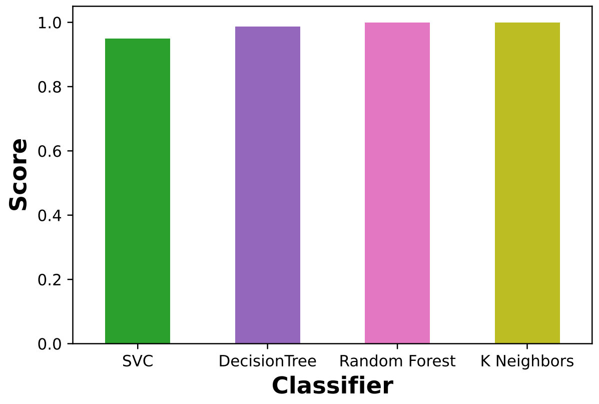


Fig. 9. The machine learning classifiers K-Nearest Neighbors, Decision Tree, Random Forest and Support Vector Machine was fitted to the training dataset (obtained from the FallDataSet by feature extraction) and the resulting accuracy scores.

accuracy, F1-score, precision, and recall is shown in figure [10]. The advantage of using a deep neural network is that these models operate directly on raw sensor data, extracting features by themselves in its convolutional layer. However, in K-NN, RF, and SVM classifiers, the data must be fed after feature extraction, i.e., data preprocessing is essential.

The final models (K-NN, LSTM, and Conv1D +LSTM) have higher accuracy than SVC and DT classifiers. The new approach of using CNN and LSTM together also improved the accuracy.

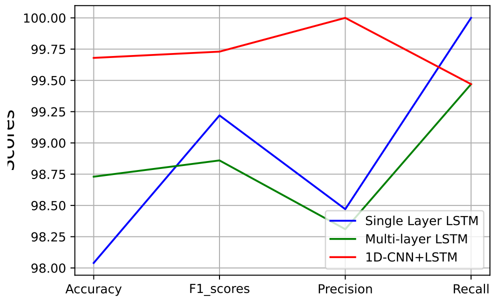


Fig. 10. The machine learning classifiers K-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machine was fitted to the training dataset (obtained from the FallDataSet by feature extraction) and the resulting accuracy scores.

TABLE I
PERFORMANCE OF THE MODEL USED FOR FALL DETECTION USING WEARABLE SENSORS.

Machine learning model	Accuracy	f1-scores	Precision	Recall
K-NN classifier (reduced features)	99.68%	-	-	-
Single layer LSTM	98.04%	99.22%	98.47%	100%
Multi-layer LSTM	98.73%	98.86%	98.31%	99.47%
Conv1D + LSTM	99.68%	99.73%	100%	99.47%

B. Comparison with other work

Machine learning-based techniques differ from each other in multiple factors—the feature set used, sensors employed, placement of sensors, algorithms applied, the dataset used, performance parameters monitored, and so on. We compared our results with some of the recent work on fall detection [5]–[7], [11]–[14] with the appropriate model used and the results are tabulated on Table [II].

VI. CONCLUSION

This project aims to address the possible solution to detect falls. We tested different machine learning and deep learning models on wearable sensor datasets to reach the proposed problem (goal). The multi-layer LSTM and the model composed of CNN and LSTM reached the highest accuracy of 100%. These results are very good in comparison to recent research on similar data. To conform to the final performance of the models, we need to do some hyper-tuning of the parameters and calculate the learning rate. Because of the time limitation, we could not perform this task. This work could be done in the future.

The data was collected for the experimental purpose from healthy volunteers. However, testing this model in real-world applications might be challenging. Therefore, future consequences of these methods' real-world application should be

TABLE II
COMPARATIVE BETWEEN OUR RESULTS AND THE OTHER STUDIES PRESENTS IN THE LITERATURE.

Contributors	Machine learning model	Results
R. Malekian, et.al(2016) [6]	kNN, ANN, and SVM	KNN:Acc=87.5%,Sens=90.70%, Spec.=83.78%
Jefiza et al. (2017) [5]	back-propagation neural (network (BPNN))	Acc= 98.182%,Sens=95.161%, Spec.=99.367%
Yu et al.(2018) [7]	Hidden Markov Model-Based Fall Detection	predictive value 0.981 and sensitivity of 0.992
Kao et al. (2017) [11]	GA-SVM, SVM classifiers	Acc= 94.1%,Sens=94.6%, Spec.=93.6%
Musci et al. (2018) [12]	RNN model with LSTM blocks	high precision on fall detection
Fakhrulddin et al.(2017) [13]	deep CNN	CNN used to identify falls but yet could not improve the robustness of fall detection
M. Galvão, et. Al(2021) [14]	multimodal approach with CNN & LSTM	a multimodal solution presents an improvement in the accuracy

addressed. In addition, elderly people might not feel comfortable with these wearable sensors attached to their body parts. The portable device running on the Linux version should be capable of running the model, and the SD card to store the sensor data is necessary. The captured data needs to inform the timestamp to sync information. We encourage future work on the application of the proposed model with real-world scenarios to evaluate it on generalizing to specific person groups such as elderly citizens.

VII. Appendix

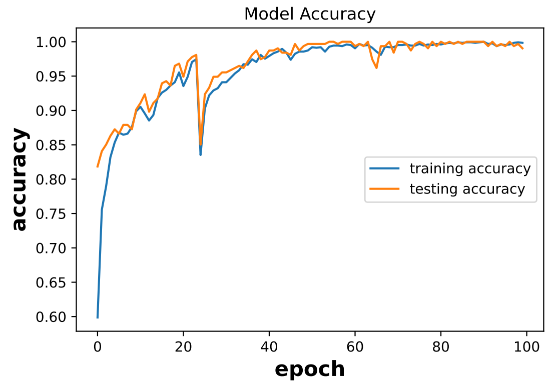
A. *Implemented codes*

The methodology delineated above was operationalized utilizing an array of computational tools and technologies. The core algorithmic implementation was performed using Tensorflow 2.8.0 and Keras 2.8.0, with Sklearn 1.0.1 employed for any auxiliary machine-learning tasks. The entire codebase was written in Python 3.8. The computational environment comprised a DELL Inspiron 15 7000, equipped with an NVidia GeForce GPU, which facilitated efficient data processing and model training. For reproducibility and wider dissemination of our work, we have made our codebase publicly accessible. It can be retrieved from our GitHub repository at https://github.com/shailendrabhandari/ACIT4630_Advanced_MLandDL.git.

B. *Visual Analysis of Model Performance during Training and Testing Stages*

All three neural networks under investigation demonstrated exemplary performance, with test accuracies exceeding 99%. This suggests that our model misclassifies fall events as non-fall activities in approximately ±0.9% of cases in the single-layer LSTM. A rapid increase in accuracy within the initial 40 epochs, as depicted in Figures [11, 12], underscores the network's swift learning capability. Following this phase, the accuracy curves plateau, indicating optimal learning without indications of overfitting for all models except the multi-layer LSTM, as illustrated in Figure [11(DOWN)].

Parallel conclusions can be drawn from the training and validation loss learning curves. For the single-layer LSTM Figure [13 (UP)] and Conv1D + LSTM models (Figure [14]), the learning curves reflect an ideal fit. An ideal fit, the target of any learning algorithm, is characterized by a consistent decrease in the loss until a point of stability is reached, with a minimal divergence between the final loss values for training and validation sets. It is anticipated that the model's loss will invariably be lower on the training dataset as compared to the validation dataset, resulting in a discrepancy between the training and validation loss learning curves, referred to as the "generalization gap". An exemplary illustration of this generalization gap is provided in Figure [14].



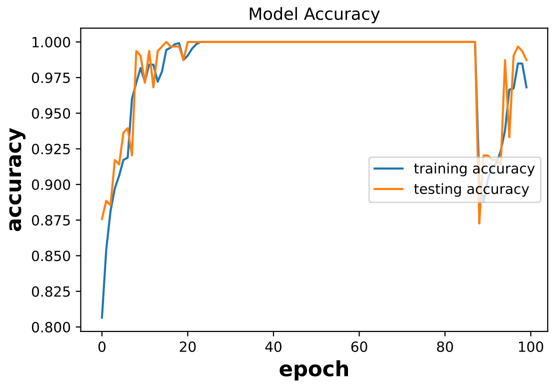


Fig. 11. Training and test accuracy plots for UP: LSTM single layer DOWN: multi-layer LSTM.

References

[1] Ponce, H., Martínez-Villaseñor, L. & Nuñez-Martínez, J. Sensor location analysis and minimal deployment for fall detection system. *IEEE Access* **8**, 166678–166691 (2020).

[2] Organization, W. H., Ageing, W. H. O. & Unit, L. C. *WHO global report on falls prevention in older age* (World Health Organization, 2008).

[3] de Quadros, T., Lazzaretti, A. E. & Schneider, F. K. A movement decomposition and machine learning-based fall detection system using wrist wearable device. *IEEE Sensors Journal* **18**, 5082–5089 (2018).

[4] Vallabh, P., Malekian, R., Ye, N. & Bogatinoska, D. C. Fall detection using machine learning algorithms. In *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–9 (2016).

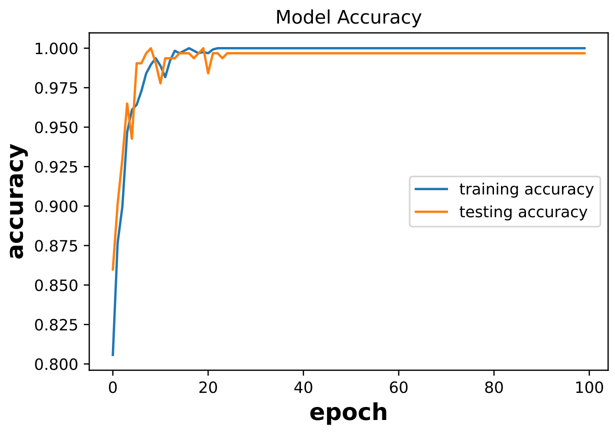


Fig. 12. Train and test accuracy curve for Conv1D + LSTM

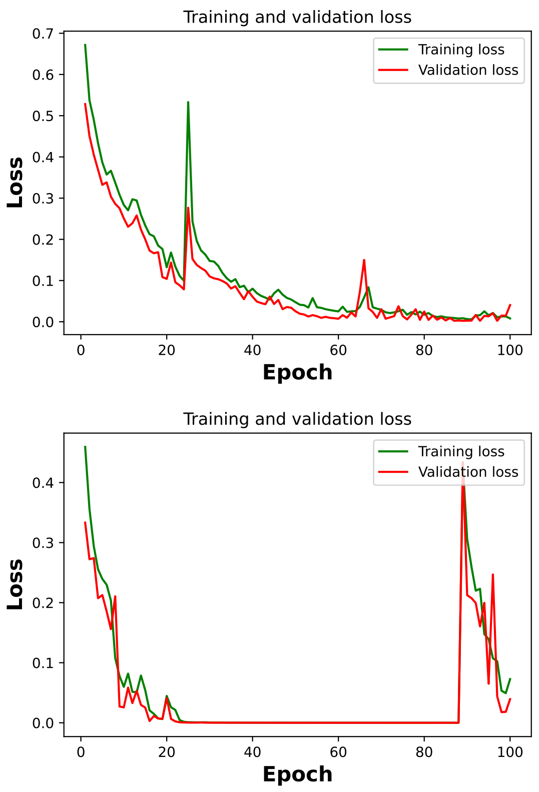


Fig. 13. Train and test learning curve for UP: LSTM single layer DOWN: multi-layer LSTM.

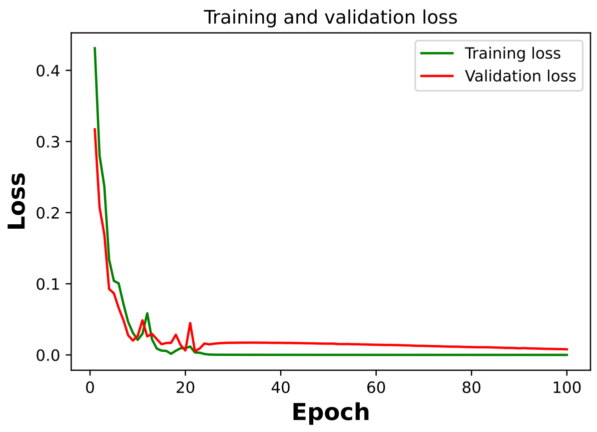


Fig. 14. Train and test learning curve for single layer LSTM.

[5] Jefiza, A., Pramunanto, E., Boedinoegroho, H. & Purnomo, M. H. Fall detection based on accelerometer and gyroscope using back propagation. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* **2017-December**, 19–21 (2017).

[6] Hossain, F., Ali, M. L., Islam, M. Z. & Mustafa, H. A direction-sensitive fall detection system using single 3d accelerometer and learning classifier. In *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*, 1–6 (IEEE, 2016).

[7] Yu, S., Chen, H. & Brown, R. A. Hidden markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring. *IEEE journal of biomedical and health informatics* **22**, 1847–1853 (2017).

[8] Chelli, A. & Pätzold, M. A machine learning approach for fall detection and daily living activity recognition. *IEEE Access* **7**, 38670–38687 (2019).

[9] Wang, H., Li, M., Li, J., Cao, J. & Wang, Z. An improved fall detection approach for elderly people based on feature weight and bayesian classification. In *2016 IEEE International Conference on Mechatronics and Automation*, 471–476 (IEEE, 2016).

[10] Genoud, D., Cuendet, V. & Torrent, J. Soft fall detection using machine learning in wearable devices. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 501–505 (2016).

[11] Kao, H.-C., Hung, J.-C. & Huang, C.-P. Ga-svm applied to the fall detection system. In *2017 International Conference on Applied System Innovation (ICASI)*, 436–439 (2017).

[12] Musci, M., Martini, D. D., Blago, N., Facchinetti, T. & Piastra, M. Online fall detection using recurrent neural networks. *CoRR* **abs/1804.04976** (2018). URL http://arxiv.org/abs/1804.04976. 1804.04976.

[13] Fakhrulddin, A. H., Fei, X. & Li, H. Convolutional neural networks (cnn) based human fall detection on body sensor networks (bsn) sensor data. In *2017 4th International Conference on Systems and Informatics (ICSAI)*, 1461–1465 (2017).

[14] Galvão, Y. M., Ferreira, J., Albuquerque, V. A., Barros, P. & Fernandes, B. J. T. A multimodal approach using deep learning for fall detection. *Expert Systems with Applications* **168**, 114226 (2021). URL https://www.sciencedirect.com/science/article/pii/S0957417420309489.

[15] Liu, K.-C. *et al.* Deep learning based signal enhancement of low-resolution accelerometer for fall detection systems (2020). URL https://arxiv.org/abs/2012.03426.

[16] Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

[17] Gao, F., Hu, G., Feng, W., Matsuo, T. & Alagar, V. Advances in mathematical methods for image and signal processing. *Journal of Applied Mathematics* **2014**, 2–4 (2014).

[18] Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **9**, 1735–1780 (1997). URL https://doi.org/10.1162/neco.1997.9.8.1735. https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf.

[19] Özdemir, A. T. & Barshan, B. Detecting Falls with Wearable Sensors Using Machine Learning Techniques. *Sensors* **14**, 10691–10708 (2014). URL https://www.mdpi.com/1424-8220/14/6/10691.