Fall Detection Using Machine Learning Algorithms

Pranesh Vallabh[1], Reza Malekian[1, *], Ning Ye[2,3], Dijana Capeska Bogatinoska[4]
[1]Department of Electrical, Electronic and Computer Engineering,
University of Pretoria, Pretoria, 0002, South Africa
[2] College of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003,China
[3]Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003,China
[4] Faculty of Computer Science and Engineering, University of Information Science and Technology "St. Paul the Apostle", Ohrid Republic of Macedonia
*Corresponding author: Dr. Reza Malekian, reza.malekian@ieee.org

Abstract— In this paper, the recognition of and the differentiation between fall activities and activities of daily living (ADL) was performed using the MobiFall dataset. A large database was constructed to train and validate the model. Feature selection methods were implemented to reduce dimensionality. Five different classification algorithms were implemented and evaluated based on their accuracy, sensitivity, and specificity achieved. The k- Nearest Neighbors' algorithm obtained an overall accuracy of 87.5% with a sensitivity of 90.70%, and a specificity of 83.78%.

Keywords—Fall detection, machine learning, public dataset, smartphone

I. INTRODUCTION

Elderly people have a higher risk of death or injury resulting from falls [1]. More than a third of the over-aged population fall each year which can result into fear and loss of independence. The increase in population of elderly people also increases the demand for healthcare systems [2].

Fall detection systems are categorized into the following three groups: ambience device, camera-based systems and wearable devices. Ambience devices are attached around an area which can detect falls using the following sensors: pressure, PIR, Doppler, microphone and accelerometer sensors. The advantages of this method is that it is cheap and non-intrusive however, disadvantages include the range and environmental factors which can result in low accuracy [2]. Computer vision makes use of cameras to track the user movements. A fall may be detected when the user is inactive for a long time [1]. Advantages of the system is that it can detect multiple events simultaneously and are less intrusive since they are not worn [2]. The disadvantages of this system are that it is limited to a specific area and does not guarantee privacy and is expensive [1], [2]. Wearable devices which are attached on the user include the following sensors: an accelerometer and a gyroscope. Advantages of these devices are that they are portable, cheap and easy to use. Disadvantages of these worn devices are that some users forget or refuse to put it on, are intrusive and can produce a lot of false alarms [1], [2].

The use of smartphones to detect fall events is a very popular method due to the decrease in the manufacturing costs of smartphones as a result of the development of inexpensive Micro Electro Mechanical System (MEMS) sensors. These sensors provide better computational capabilities when compared to other wearable devices [2], [3], [4]. Smartphones include the following sensors: accelerometer, gyroscope, compass, magnetic field, proximity light and pressure sensors. This allows different algorithms to be applied to increase the accuracy of any data required [5].

The accuracy of the system depends on the sensors used and the type of classifications. Two considerations that are taken into account for fall detection systems are the speed and the accuracy of the fall detection classifier. Factors that influence the identification and classification of fall activities and ADLs include environment, difference in movements, and the displacement of the sensor [6]. Classification methods can be divided into two models, supervised and unsupervised models. In a supervised model, the outputs can be controlled and can be easily model when compared to the unsupervised model [4]. Unsupervised models that have been used in fall detection systems such as one-class support vector machine [7]. Supervised models include support vector machine (SVM), decision threshold tree, Naive Bayes, least squares method (LSM), k-Nearest Neighbor (k-NN), and artificial neural networks (ANNs). Supervised models are implemented in this paper.

An accelerometer, gyroscope and magnetometer were used to collect body movements which were classified using a threshold decision tree [8]. The E-FallD system uses an accelerometer and users' personal information such as height, weight, sex and age is used to adjust the thresholds to make it suitable for anyone who would make use of this system. The system uses a threshold decision tree and includes a button to cancel if a false alarm is triggered [2]. Based on related work, a threshold decision tree is the most implemented algorithm in fall detection monitoring applications; the reason being that it is computationally fast and is energy efficient due to the low computational complexity [8], [9]. Threshold detection algorithm is not viable since the thresholds are different for each and different activities have different thresholds which are not taken into consideration [9]. A tri-axis accelerometer sensor was used with a hidden Markov model (HMM) based on the acceleration time series [10]. The magnitude of both the accelerometer and gyroscope sensors were extracted and fed into a k-NN classifier [11]. Naive Bayes achieved the highest accuracy when compared to the rest of the classifiers, where an accelerometer was used to extract data movements [12]. Features were extracted using k-means and were fed into a two-level hybrid classification algorithm which combines neural network and softmax regression [13]. Six accelerometers were used to model user movements, where statistical features were extracted and fed into different classifiers. The k-NN and LSM were the most reliable system [14].

Fall detection classification systems can be characterized through five stages: pre-processing, feature extraction, feature selection, model training and classification. When these steps are combined together into the proposed system, a generic model can be shown in figure 1. Advantages of the proposed classification system is that the feature extraction and selection techniques are used to optimize the classification algorithm. The developed classifiers can become too complex and result in long computational periods which can suffer from high biasing [4]. It is important to select and minimize the features used as inputs to the classifier [4]. Most of the studies do not include a feature selection method to decide whether the feature extracted contain information require to differentiate between ADL and fall activities. Previous work [15] has been done in fall detection system, in this paper new machine learning algorithms and feature extraction methods will be investigated.

The aim of this paper is to implement different machine learning methods, based on the smartphone dataset, to determine the performance of the system based on features selected from the extracted features. Various machine learning algorithms, excluding threshold decision tree, will be evaluated in order to determine which algorithm performs the best overall and will be compared to the threshold decision trees in [16]. The following classification methods were implemented: neural network, support vector machine (SVM), k- Nearest Neighbors (k-NN), least squares method (LSM), and Naive Bayes. The overall objective is to successfully classify both ADL and fall activities; and determine which machine learning algorithm is to be implemented or to replace the common threshold decision tree. The contribution in the paper is the feature selection method which uses a rank-based system to compute the t-test on each feature; and uses the p-value to determine which features to use in the classifier are. As per my knowledge, this feature selection method has not been implemented for this application. The paper provides results of the different machine learning algorithms using a public dataset, to determine the best classifier.

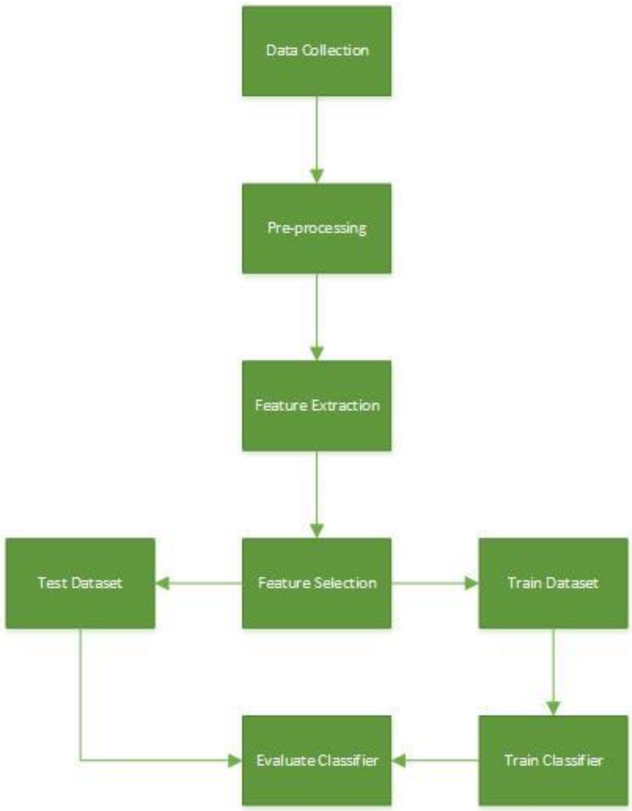

Fig. 1. Flow diagram of the proposed machine learning based fall detection classification system.

II. DESIGN IMPLEMENTATION

A. Data Collection

The "MobiFall" dataset is used for fall detection which is downloaded from the website of the Biomedical Informatics and eHealth Laboratory at the Technological Educational Institute of Crete[1] [16]. Data captured from the "MobiFall" dataset through a smartphone which was placed in the user's trouser pocket. Table I shows what smartphone sensors were

[1] www.bmi.teicrete.gr

used and what data measurements were recorded in the “MobiFall” dataset by what smartphone sensor.

TABLE I. MEASURMENTS RECORDED IN “MobiFall” [16]

Sensor	**Measurements recorded in “MobiFall” dataset**
Accelerometer	Acceleration force along the x, y, and z axis (m/s^2).
Gyroscope	Rate of rotation around the x, y, and z axis (rad/sec).
Orientation (software based)	Angle around the z (azimuth), y (pitch) and x (roll) axis in degrees.

From Table I, the orientation sensor data was calculated instead of reading it from a hardware component. The orientation sensor data is calculated using both the accelerometer and geomagnetic field sensor [16]. A low cost gyroscope suffers from time-varying zero shift seriously which can cause significant errors to the calculated angular acceleration and angular position, which is calculated using differential and integral operations. The errors can be compensated for by using a magnetograph, but its computation is too complex and does not meet the specifications of a real-time system [10]. The use of the orientation senor data resulted in a low performance system shown in [16]. Due to the gyroscope’s error and the orientation sensor’s low performance, the accelerometer sensor will be used to classify fall activities from ADLs. From Table I, it can be seen that a tri-axis accelerometer was used.

Two types of activities were captured, namely fall activities and ADL’s. The ADL’s activities are captured by the “MobiFall” dataset which include the following properties: standing, walking, jogging, jumping, walking up the stairs, walking down the stairs, and sitting on a chair. The different “fall” activities which was captured by the “MobiFall” dataset is shown in Table II.

The raw accelerometer data that is extracted from the dataset lies in a range of [-20, 20] m/s^2. The raw tri-accelerometer axis value of the ADL’s is shown in figure 2.

TABLE II. FALL ACTIVITIES FROM “MobiFall” [16]

Activity	**Description**
Forward lying	Falling forward from standing and using hands to reduce the impact of the fall.
Front knees lying	Falling forward from standing with the knees as the first impact.
Back sitting chair	Falling backward when trying to sit on the chair.
Sideward lying	Falling sideward with legs being bend.

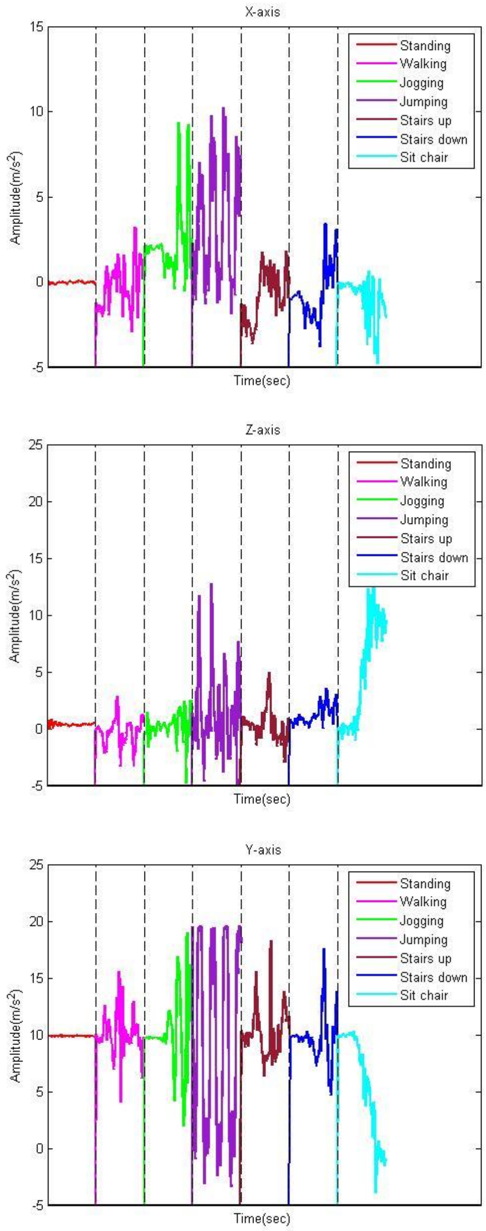

Fig. 2. Raw accelerometer data of the x,y and z axes for ADLs.

The raw tri-accelerometer axis value of the fall activities is shown in figure 3.

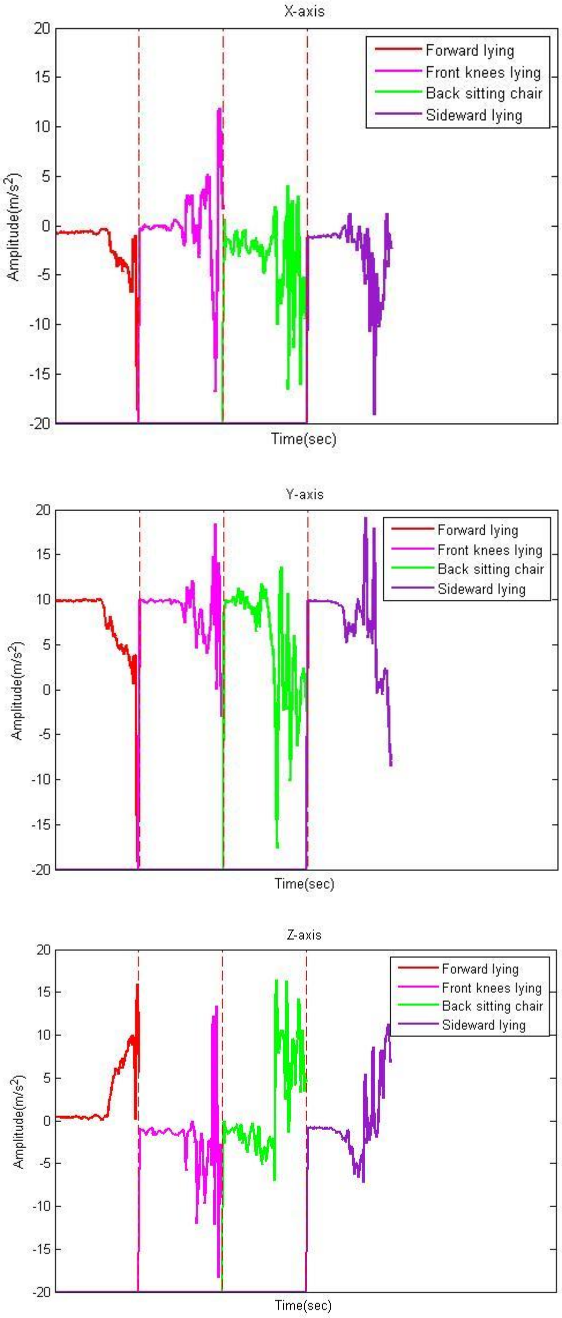

Fig. 3. Raw accelerometer data of the x, y and z axes for fall activities.

B. Pre- Processing

This process involves acquiring the data from the dataset and creating a larger dataset that will be used for training and testing. The data from the different sensors, available on the smartphone, are not sampled at a constant sampling rate but rather using "SENSOR_DELAY_FASTEST" which gathers data at the highest possible sampling rate [16]. The authors in [16] stored the timestamp to calculate a constant sampling rate. In [17] the sampling rate was calculated to be 100 Hz. To validate sampling rate, the following was formula was used,

$$sampling\ rate = \frac{1}{Timestamp(i+1)-Timestamp(i)}, \qquad (1)$$

where $i \in [1, number\ of\ acceleration\ records]$. Using equation (1) the sampling rate is calculated to be around 100 Hz. To achieve a constant sampling rate a timer can be used.

Accelerometer data will be pre-processed to remove the effects of several sources that influence the sensor, this is shown in Table III.

TABLE III. SOURCES WHICH INFLUENCE THE ACCLEROMETER [1]

Source	**Description**
1	Acceleration due to body movements.
2	Gravitational acceleration.
3	External vibrations e.g. from vehicles.
4	Accelerations due to accelerometer bouncing against other objects which produces mechanical resonance.

Sources 1 and 2 relate to body movements, whereas source 3 and 4 add noise to the accelerometer's output. A 3rd order median filter is implemented to attenuate the noise produced by source 3 and 4 [1]. The median filter was computed using the Matlab toolbox function "medfilt1" [18]. To remove the acceleration due to body movement (source 1) from an accelerometer, a low pass filter was used which is given by [19],

$$\begin{aligned} A'_y(t_{i+1}) &= \alpha \times A'_y(t_i) + (1-\propto) \\ &\quad \times A_y(t_{i+1}) \\ \Delta A'_y(t_i) &= A'_y(t_{i+1}) - A'_y(t_i) \end{aligned} \qquad (2)$$

where $A_y(t_{i+1})$ and $A'_y(t_i)$ are the sampling and smooth values with α set to 0.5 [19].

The smoothing effect of the filters is shown in figure 4, where the noise sources is removed from the accelerometer z-axis. The data of all the activities are combined into one dataset and shuffled, to reduce biasing. The data in the large dataset is split into training and test data. The system uses the 75% of the dataset for training and rest for testing the accuracy of the system.

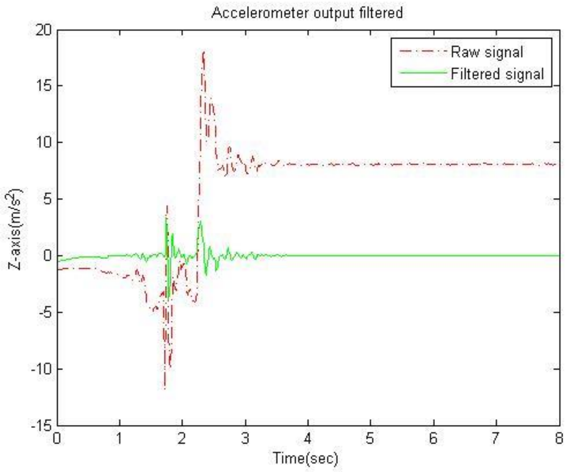

Fig. 4. The raw and the filtered signal from the z-axis axis after the median and low pass filter was applied to it.

C. Feature Extraction

Feature extraction plays an important role in a fall detection classification system, since the selected features will determine the accuracy of the system. Feature extraction is usually performed on data obtained from a window interval. The window allows for the creation of a set of values from which a unique feature can be extracted. To create a window interval, the time index of the maximum signal magnitude vector (SMV) peak is first found in each activity record [14]. The SMV describes the changes in human movement and detection of a possible fall which is given by [14], [11],

$$SMV = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}, \tag{3}$$

where x(t), y(t), z(t) represent the readings of the x, y and z axis' of the accelerometer [14]. The maximum peak is then used as a reference point to create a window interval [14]. The size of the window is important and according to various studies, it should contain at least one cycle of a single activity to uniquely identify it, a window size can range from 1 to 7.5 second [4], [10]. A window interval of 4s was chosen, which was found to be optimal in related work [14]; where 2s is before the maximum SMV peak, and the other 2s is after the SMV peak. This is done to allow to capture the fall data from when the impact started and when it settled down [14].

Most features are derived or calculated based on statistical models, therefore include values such as mean, median, max, min, standard deviation, etc. [4], [14]. Features that are extracted can be either time or frequency based features, both features can improve the accuracy of the model [4]. The features extracted from the accelerometer tri-axis derived from related work in [4] and [14]; which are calculated with Matlab's in-built functions which are given in Table IV.

TABLE IV. FEATURES EXTRACTED FROM “MobiFall”

Time Domain Features	**Frequency Domain Features**	**SMV Features**
Mean	Mean	Skewness
Maximum	Maximum	Kurtosis
Minimum	Minimum	
Standard deviation	Standard deviation	
Interquartile Range (IQR)	Skewness	
Median	Kurtosis	

From Table IV, the kurtosis and skewness is calculated using the following formulas which is given by [14],

$$skewness(s) = \frac{1}{N\sigma^3}\sum_{n=1}^{N}(s_n - \mu)^3, \tag{4}$$

$$kurtosis(s) = \frac{1}{N\sigma^4}\sum_{n=1}^{N}(s_n - \mu)^4, \tag{5}$$

where s represents the data and N the size of the data [14].

The total number of features that were extracted was 38 where 18 features were obtained from time-domain (6 features x 3 axis of accelerometer), other 18 features from frequency-domain (6 features x 3 axis of accelerometer), and 2 features from SMV data. The features were extracted in the window interval. All of the features were normalized in the range between [0, 1] using the unity-based formula. Normalization allows all data to be treated the same and reduces biasing and variance [4].

D. Feature Selection

The performance of a classification algorithm directly depends on the size of the features extracted. To improve the performance of the system, features should be minimize and the effects of dimensionality should be reduced [4]. One way to reduce dimensionality would be to apply feature selection algorithms. Feature selection is also used to determine which features are informative and can significantly differentiate between ADL and fall activities. To determine which features to select from, the features are extracted and a rank based system was implemented. A filter approach rank based system was implemented which computes the t-test on each feature and the p-value of the test is used to determine which features to be used. The "ttest2" Matlab function from the toolbox is used to get the p-values [20]. An empirical cumulative distribution function (CDF) of the p-values is shown in figure 5 which shows the difference in feature values between the ADL and fall activities.

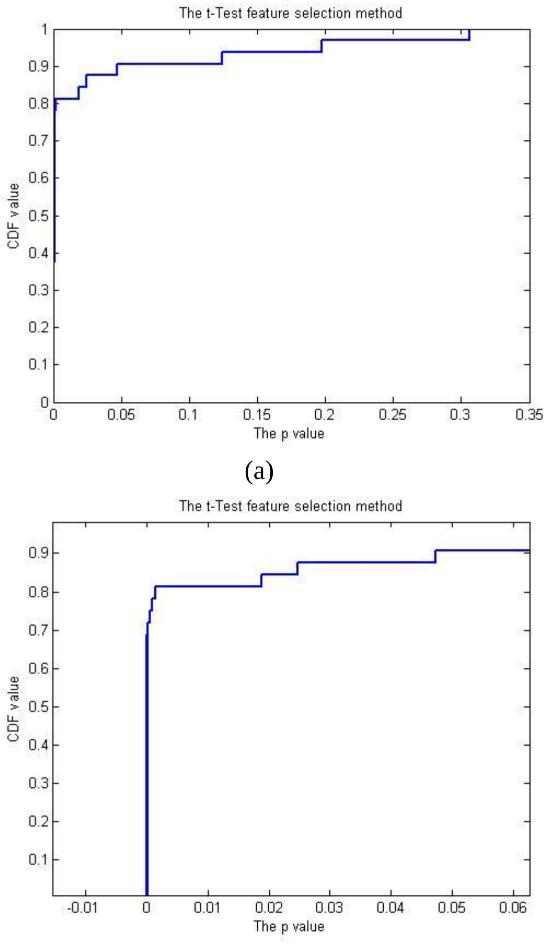

Fig. 5. (a) All the p-values and (b) the p-values from [0, 0.5].

Features that are selected for classification mainly use a p-value range of [0.001, 0.05] [20]. The five features that will be used to train the different classifier are given in Table V with its corresponding p-value output.

TABLE V. TOP FIVE FEATURES SELECTED

Feature number	**Features**	**Type of features**	**p-value**
1	Z median	Time	0.0026
2	X mean	Frequency	0.0050
3	Y mean	Frequency	0.0071
4	X median	Time	0.0081
5	Skewness	SMV	0.0181

1) Naive Bayes

Naive Bayes is a probabilistic model which is based on Bayes Theorem. The prior probabilities are used to classify new features. Naive Bayes was modeled as a normal function, and was implemented in Matlab [12], [21].

2) The Least Squares Methos (LSM)

Two average reference feature vectors are calculated for the ADL and fall activities and a new feature vector is compared to the reference vector using the following equation [14],

$$\varepsilon_i^2 = \sum_{m=1}^{M} (x_m - r_{im})^2, \qquad (6)$$

where x_m is the new feature vector and r_{im} is the average reference vectors with i = 1, 2. To determine what class the new feature belongs to; ε_i^2 is minimized to classify the new feature [14].

3) Artifical Neural Newtork (ANN)

The ANN is a two-layer feed forward network which consists of sigmoid hidden and output neurons. The ANN was implemented using the "nnstart" toolbox in Matlab. The system is trained using the scaled conjugate gradient backpropagation [12], [14], [22].

4) Support Vector Machine

The SVM was implemented in Matlab using libsvm toolbox [22]. The following functions were used "svmtrain" to train the system and "svmpredict" to test the accuracy of the system. The type of SVM system implemented is a C – Support vector. The radial basis kernel was implemented.

III. RESULTS AND DISCUSSION

A total of 38 features were extracted and a filter rank based system was used to eliminate features with no information. The top five features were used in the classification model which provide decent results. The classification model can produce the following four possible outcomes [3]:

1. True Positive (TP) when a system properly detects a fall when fall has occurred.
2. False Positive (FP) when a system detects a fall when no fall has occurred.

3. True Negative (TN) when a system detects no fall when no fall has occurred.
4. False Negative (FN) when a system detects no fall when a fall has occurred.

The performance indicators that were used to access the system are sensitivity, specificity and accuracy. Sensitivity represents the capacity to detect falls which is given by [3],

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100. \qquad (7)$$

Specificity represents the capacity to only detect falls and ignore non- fall events which is given by [3],

$$\text{Specificity} = \frac{TN}{FP + TN} \times 100\,. \qquad (8)$$

Accuracy represents the portion of true results among the population which is given by [3],

$$\text{Accuracy} = \frac{TN + TP}{TP + TN + FP + FN} \times 100\,. \qquad (9)$$

Table VI shows a summary of how each classifier performed based on the performance indicators.

TABLE VI. SUMMARY OF CLASSIFIER PERFORMACE

Classifier	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
Naives Bayes	80.0	85.11	72.73
k-NN	87.5	90.70	83.78
LSM	75.43	77.26	74.51
ANN	85.87	89.23	81.43
SVM	86.75	89.74	82.93

Overall k-NN, ANN, SVM had the best accuracy compare to LSM and Naïve Bayes. ROC (Receiver Operating Characteristic) curves where the classifier performance is plotted as the discrimination threshold varies. The discrimination threshold is used to determine the class where the instance belongs to [20]. ROC curves for each classifier are shown in figure 6.

The ROC curve shows the performance of each classifier. The best classifier that can be seen from the ROC curve is the k-NN with an accuracy of 87.5%, sensitivity of 90.70%, and a specificity of 83.78%. In [16] three different types of threshold decision tree were implemented using the "MobiFall" dataset. The best threshold decision tree achieved a sensitivity of 0.71 and specificity of 0.84. The five classifiers that were implemented, excluding LSM, performed far better than the threshold decision tree. This shows that although threshold decision tree is computationally fast, they do not provide an accurate way of detecting falls. Table VII shows a comparison of the top three classifier results obtained in this research to related works.

From table VII, ANN and SVM perform better than the related work. The accuracy of k-NN is lower compare to the related work; since the related work [14] used six accelerometer sensors to capture the user movements compared to the "MobiFall" dataset which used only one sensor. Factors that influence the accuracy is the placement of sensors on the body. It is found that the device at upper trunk of the body, below the neck and above the neck provides the best results, reason being that the other locations contain high movement frequency and complexity [10]. The "MobiFall" dataset was recorded in user's pocket, which is below the waist. Another factor that may result in an inaccurate model is the change in smartphone reference or orientation [4]. From the classifiers implemented, the k-NN, ANN, and SVM are viable options for implementation.

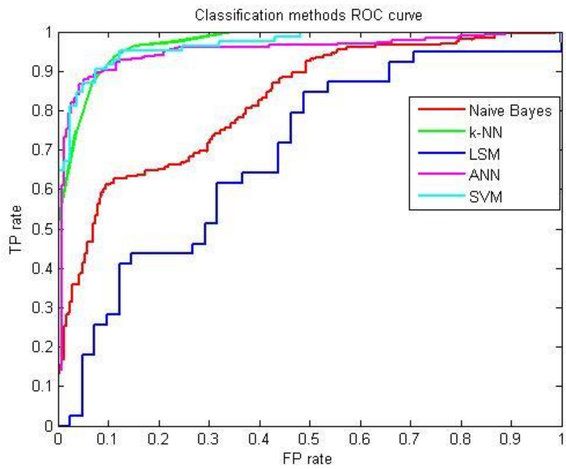

Fig. 6. ROC curves for different classification methods.

TABLE VII. COMPARISON OF RESULTS FOR VERIFICATION

Classifier	**Classification results (%)**	**Related work (%)**	**Reference**
k-NN	87.5	99.91	[14]
ANN	85.87	84.57	[12]
SVM	86.75	64.40	[12]

IV. CONCLUSIONS

In this paper, different classification methods were implemented for fall detection and they were able to distinguish between ADL and fall activities in the "MobiFall" dataset. The data was pre-processed by a median and low-pass filter. A filter rank based system was implemented to extract the top five features in order to optimize algorithm's dimensionality. The five classification methods that were implemented are: Navie Bayes, k-NN, ANN, SVM, and LSM. The k-NN with k equal to 5 achieved an 87.5% accuracy which resulted in the highest accuracy when compared to the other classification methods, and achieved better results when compared to the threshold decision tree. Possible improvements include trying to increase the accuracy of the system and the inclusion of more ADLs to make the system more applicable to real life scenarios. For future research, creation of a new dataset that is recorded on the upper torso and investigation into migrating the classifier method to a real–time model. Also, trying to make the algorithm user independent without including more test subjects.

REFERENCES

[1] Yi He, Ye Li and Shu-Di Bao, "Fall detection by built-in tri-accelerometer of smartphone", in *Biomedical and Health Informatics, Proceedings of 2012 IEEE-EMBS International Conference*, Jan. 5-7,2012, pp. 184-187.

[2] Y. Cao, Y. Yang and W. Liu, "E-FallD: A fall detection system using android-based smartphone", in *Fuzzy Systems and Knowledge Discovery, 2012 9th International Conference*, May 29-31, 2012, pp. 1509-1513.

[3] B. Aguiar, T. Rocha, J. Silva and I. Sousa, "Accelerometer-based fall detection for smartphones", in *Medical Measurements and Applications, 2014 IEEE International Symposium*, June 11-12, 2014, pp. 1-6.

[4] J. Wannenburg and R. Malekian, "Physical Activity Recognition From Smartphone Accelerometer Data for User Context Awareness Sensing", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-8, May 2016.

[5] E. Thammasat and J. Chaicharn, "A simply fall-detection algorithm using accelerometers on a smartphone", *The 5th 2012 Biomedical Engineering International Conference*, Dec. 5-7, 2012, pp. 1-4.

[6]T. Bennett, J. Wu, N. Kehtarnavaz and R. Jafari, "Inertial Measurement Unit-Based Wearable Computers for Assisted Living Applications: A signal processing perspective", *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 28-35, Mar. 2016.

[7] Miao Yu, Yuanzhang Yu, A. Rhuma, S. Naqvi, Liang Wang and J. Chambers, "An Online One Class Support Vector Machine-Based Person-Specific Fall Detection System for Monitoring an Elderly Individual in a Room Environment", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1002-1014, Nov. 2013.

[8] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini and S. Valenti, "A High Reliability Wearable Device for Elderly Fall Detection", *IEEE Sensors J.*, vol. 15, no. 8, pp. 4544-4553, Apr. 2015.

[9] Y.Charlon, N. Fourty, W. Bourennane, and E. Campo, "Design and evaluation of a device worn for fall detection and localization: Application for the continuous monitoring of risks incurred by dependents in an Alzheimer's care unit", *Expert Systems with Applications*, vol. 40, no. 18, pp. 7316-7330, Dec. 2013.

[10] Lina Tong, Quanjun Song, Yunjian Ge and Ming Liu, "HMM-Based Human Fall Detection and Prediction Method Using Tri-Axial Accelerometer", *IEEE Sensors J.*, vol. 13, no. 5, pp. 1849-1856, Feb. 2013.

[11] H. Jian and H. Chen, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine", *China Communications*, vol. 12, no. 4, pp. 23-31, June 2015.

[12] X. Yang, A. Dinh and L. Chen, "A wearable real-time fall detector based on Naive Bayes classifier", in *CCECE, 2010 23rd Canadian Conference,* May 2-5, 2010, pp. 1-4.

[13] S. Abeyruwan, D. Sarkar, F. Sikder and U. Visser, "Semi-Automatic Extraction of Training Examples From Sensor Readings for Fall Detection and Posture Monitoring", *IEEE Sensors J.*, vol. 16, no. 13, pp. 5406-5415, Apr. 2016.

[14] A. Özdemir and B. Barshan, "Detecting Falls with Wearable Sensors Using Machine Learning Techniques", *Sensors*, vol. 14, no. 6, pp. 10691-10708, June 2014.

[15] P. Vallabh, R. Malekian, N. Ye, D.C. Bogatinoska, A. Karadimce, J. Ritonja, "Classification of fall detection in elderly persons based on smart phone data,", *Journal of Biotechnology*, vol. 231, pp. s29-s30, Aug. 2016.

[16] G. Vavoulas, M. Pediaditis, E. Spanakis and M. Tsiknakis, "The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones", in *BioInformatics and BioEngineering, 13th IEEE International Conference*, Nov. 10-13, 2013, pp. 1-4.

[17] R. Igual, C. Medrano, I. Plaza, "A comparison of public datasets for acceleration-based fall detection," *Medical Engineering and Physics*, vol. 37, no. 9, pp. 870-878, Sept. 2015.

[18]"1-D median filtering - MATLAB medfilt1", *Mathworks.com*, 2016. [Online]. Available: http://www.mathworks.com/help/signal/ref/medfilt1.html. [Accessed: 21- Apr- 2016].

[19] M. Sie and S. Lo, "The design of a smartphone-based fall detection system", in *Networking, Sensing and Control, 2015 IEEE 12th International Conference*, Apr. 9-11, 2015, pp. 456-461.

[20] "Selecting Features for Classifying High-dimensional Data - MATLAB & Simulink Example", *Mathworks.com*, 2016.[Online].Available:http://www.mathworks.com/help/stats/examples/selecting-features-for-classifying-high-dimensional-data.html. [Accessed: 21- Apr- 2016].

[21]Y. Delahoz and M. Labrador, "Survey on Fall Detection and Fall Prevention Using Wearable and External Sensors", Sensors, vol. 14, no. 10, pp. 19806-19842, Oct. 2014.

[22] "Classify Patterns with a Neural Network - MATLAB & Simulink", *Mathworks.com*, 2016. [Online]. Available: http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html. [Accessed: 21- Apr- 2016].

[23] C. Chang and C. Lin, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1--27:27, 2011.Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm