Advances in Parallel Computing Algorithms, Tools and Paradigms
D.J. Hemanth et al. (Eds.)

doi:10.3233/APC220003

Development of Novel Ensembled Boosting Model (EBM) for Fall Detection

A.JackulinMahariba[a], R.AnnieUthra[a,1], Chandramouleesvar.V[b] and Swetha M E[b]
[a]*Department of Computational Intelligence, SRMIST, kattankulathur*
[b]*Department of Networking and communication, SRMIST, Kattankulathur*

Abstract. Mountain biking is an extreme sport with unpredictable terrain and several dangerous risks associated with it. Even the soundest minds might need external stimuli to alert them to be more careful at a particular moment of a potential fall. The proposed work involves developing an algorithm capable of detecting falls in mountain biking activity. Machine Learning classifier algorithms are used for fall detection. The existing fall detection algorithms are used to detect falls in environments with limited movements. Fall detection through the use of cameras causes invasion of privacy and is done in fixed environments with predictable dangers, another type is through sensors attached to the human body which acts as an obstruction to the activities of the person. The proposed Ensembled Boosting Model (EBM) classifier involves overcoming these pitfalls and developing a high accurate system to detect falls in open and unpredictable environments. The algorithm proposed in this paper aims to detect falls through real-time data such as acceleration, gyroscopic values, for any user. In the future, this algorithm can be used as a precursor to implement a real-time fall prediction device to be used by anyone and in any environment.

Keywords. Boosting, Classifiers, fall detection, Ensemblelearning.

1. Introduction

One of the most widespread outdoor recreational activities, which is a thrilling adventure sport, is Mountain Biking. This activity is physically exacting and is competed in remote terrains and sporadic weather settings, possibly leading to faintness, heat stress, dehydration, skin problems, boils, insect bites, or hypothermia. In a survey attended by about 4000 athletes, around 8000 single abrasions were reported by approximately 3500 athletes. Even though most injuries from mountain biking are slight ones, studies showed that the bikers had an average injury risk rate of 0.6% annually and 1 at least injury for 1000 hours of riding, consequently increasing the probability of getting seriously injured.
The prominent factors of risk are slippery surfaces, extreme speed, cyclists' poor judgment, and personal factors pertinent to an individual that could be prevented by safety protocols.

[1] A.JackulinMahariba[a], Department of Computational Intelligence, SRMIST, kattankulathur. Chennai. E.Mail: jackulinumar89@gmail.com

Within the sum of all injuries, 3/4th of them were slight ones, such as small grazes and simple bruises, while 1/7th of them were caused by collisions with a component of the bike; 1/10th were so serious that they required hospitalization. Fall injury treatments and recovery from it are extremely expensive and grueling (Gaulrapp, Weber, & Rosemeyer, 2000)[1]. Most mountain biking terrains are not monitored and calls for help might not be heard in isolated environments. Even a small wound might become fatal if not attended to in time. Hence, fall detection can be very useful in these challenging and dangerous situations.The purpose is to design a fall detection system for mountain bikers using factors such as acceleration, elevation, angle of tilt and such, to alert the preconfigured emergency contact within seconds of fall detection and help prevent any major damage to the person.

2. Literature Review

Many kinds of research have been done using several methods to understand balancing characteristics of humans and vehicles towards detecting falls. From the references collected throughout the past decade, the trend of creation of fall detection techniques, developing a better architecture than the previous methods to even systems to predict falls have been observed. (Castro, Delgado-Escaño, Cózar, Marín-Jiménez, Guil, & Casilari, 2019)[2] Developed a methodology using accelerometer readings that not only detects falls but also identifies people. It is a dataset independent model with a fall detection accuracy greater than 98%, subject identification of 79.6%, and a false-positive rate of less than 1.6%.Algorithms like CNN, RNN, and LSTM are used to achieve this. [3-6]Methods used by (Chen, Li, Zhang, Tian, & Chen, 2019), (Torti, et al., 2019), (Casilari-Pérez & García-Lagos, 2019) and (Alarifi & Alwadain, 2020)were similar, they used an Artificial Neural Network-based wearable system to detect falls. Accelerometer and gyroscopic values were used as features.

Apart from high performance, the method also had shortcomings such as high-power consumption, extensive use of resources, and limited configuration of sensors. (Wu, Su, Feng, Yu, & Zang, 2019)Also used a wearable sensor-based system to detect falls using Fisher discriminant analysis[7]. It is a pre-impact fall detection system that employs multiple sensors to attain a Sensitivity of 95.5% and a Specificity of 97.3%. Only forward and backward falls were considered which had long average lead times of 404ms and 376ms respectively.[8][9] Fog-based infrastructure were utilized by (Sarabia-Jácome, Usach, Palau, & Esteve, 2020)and (V & R, 2020). A fog-based DL fall detection system (Sarabia-Jácome, Usach, Palau, & Esteve, 2020) was devised that uses a wearable XYZ-axial accelerometer and GRU and LSTM architecture. The only limitation is that memory occupancy is high due to the use of high-end technologies[10] Authors like (Nooruddin, Islam, & Sharna, 2019) and (Farahbakhsh, Mozaffari, Rezazadeh, Farahbakhsh,, Yazdani,, & Sandrasegaran, 2019) came up with purely IOT-based techniques for fall detection. A survey on different fall detection methods based on IOT used devices like Raspberry Pi, Arduino, and NodeMcu in their architecture to create an effective fall detection system. The system gave an accuracy of 99.7%, a specificity of 99.6%, and a sensitivity of 96.3%.A comparison is provided between different public datasets for fall detection (Igual, Medrano, & Plaza, 2015)[12]. Famous datasets like DLR, MobiFall, and tFall were used and trained with machine learning algorithms like SVM and nearest neighbors. Different single classifier algorithms like KNN, ANN, PPCA, LDA, and RB and multi classifier algorithms are utilized (Gibson, Amira, Ramzan, Casaseca-de-la-Higuera,, & Pervez, 2016) to perform fall detection[13]. Voting Machine obtains recall of 99.0%, precision of 92.6%, specificity of 91.0%, accuracy of 95.3% and F- value of 95.7% for fall detection. CVM and CM showed similar results with CM having an accuracy of 99%. (Kattukkaran, George, & Haridas, 2017), (Baramy, Singh, Jadhav, Javir, & Tarleka, 2016) and (P, T, S, & V, 2014) developed similar automatic accident detection systems[14-16]. All these methods had a microcontroller unit and mobile phone application.

3. Dataset Description

The dataset was collected from Kaggle, and the name of the dataset is *Simulated Falls and Daily Living Activities.* The dataset contains time-series values from 6 Sensors which include 3 axis Accelerometer, Gyroscope, and a Magnetometer. They were attached to healthy and fully able people. These values were recorded during fall occurrences. 20 different types of fall values are present in the dataset such as vertical fall on the floor, fall with quick recovery, slow recovery, backward fall, etc. This is unsupervised data that was converted to supervised data by labeling each record manually. Supervised data contains 21 different target classes, 20 falls and all non-fall is classified as one class. It contains a total of 22 columns including the target column and 4 lakh records.

4. Proposed Methodology

The architecture of the fall detection is – Dataset used for training the classifiers is a benchmarked dataset. The exploratory data analysis of the dataset is completed to understand the nature of the dataset. In this stage, the different features available in the dataset and all the different classes in the target column are extracted. This phase is important for model selection and feature normalization. The data may contain features like accelerometer, gyroscopic, and elevation above ground.This data is processed (data cleaning, transforming, and validating) first and then split into training and testing data. The training data is used to develop the model to learn to detect falls. The performance of the model is assessed and experimented with different algorithms until high accuracy is achieved. All the selected classifiers use 70:30 train test split. Once the model is trained, it is then validated with testing data. Once the algorithm is producing the expected accuracy, then it is used for Real-time fall detection. The algorithm keeps running with continuous real-time data input until a fall is detected. Once a fall is detected, it goes to the final state of fall detection and reports that a fall is detected.

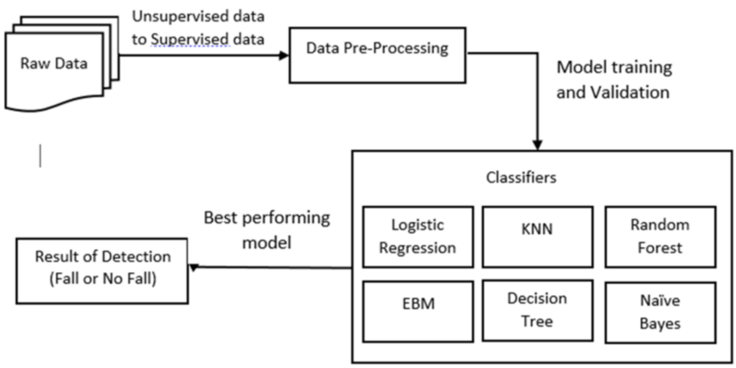

Figure 1. Architecture diagram of Fall detection system

The overall methodology can be split into 3 modules Input, Processing, and Output. The input module consists of sensors readings and data pre-processing. The sensors continuously sense the inertial measurement value with respect X, Y, and Z axis, which is resulted in a time-series data. This data is then processed and converted single records of every second before it is sent to the Fall Detection System. The processing module is the main module where the actual fall detection analysis takes place. This module decides whether a fall has occurred or not. The fall detection is already trained with training data and is tested with the real-time data generated by the sensor. The result of the classifier module is sent to the output module. Based on the result that is sent, appropriate action is taken. The two outcomes are fall detected and fall not detected.

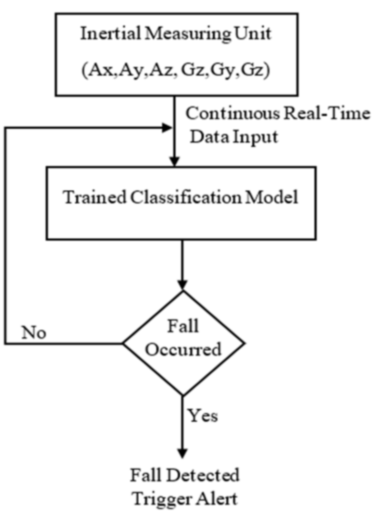

Figure 2. Flow diagram of proposed fall detection model

4.1 Classifiers

- *Logistic Regression*

It is a Supervised Learning technique and is used for predicting the categorical variable using a given set of features. For example, 0 or 1, True or False, Yes or Not, etc. It outputs probabilistic values in the range of 0 to 1. Logistic regression is similar to linear regression, but the only dissimilarity between both is that problems with continuous values are dealt with by linear regression and logistic regression helps with classification problems and uses a complex cost function called 'Sigmoid Function'. It is a simple and significant algorithm that can deliver probabilities and categorize new data using continuous and discrete datasets.

$$sigmoid(z) = \frac{1}{1+e^{-(z)}} \qquad Eq.\ (1)$$

- *K-Nearest neighbours (KNN)*

It is also called the Lazy Learner algorithm. KNN groups all the data points class-wise and classifies the new data point based on a similarity measure. Distance metrics is used to measure the similarity. The instance is assigned to the class from which it has the least distance. It is a simple and a good classification algorithm if the number of samples is large but at the same time choosing k might be tricky and it takes more time for classification. It takes more time as it needs to calculate and compare the distance of the new data with all other data points.
Distance metrics used are given in the Eqs. (2) and (3)

$$Manhattan\ Distance = \sum |x_i - y_i| \qquad Eq.\ (2)$$

$$Euclidean\ Distance = \sqrt{\sum |x_i - y_i|^2} \qquad Eq.\ (3)$$

- *Decision Tree Classifier*

It is Supervised Machine Learning that works similarly to an if-else statement where the data is split according to a condition, or the best fit tabulated by itself based on the discretion of the dataset. The whole data is represented as a tree structure where the features are present in parent nodes and the leaf nodes give the class labels. The parameters are decided based on how the data is split. Each record goes through the parameters present in each edge and gets classified into their respective class labels accordingly. The Decision tree algorithm classify unknown records in a short period and are very easy to configure.

- *Random Forest Classifier*

It is a highly flexible and user-friendly algorithm that can be used for both classifications as well as regression problems. The forest consists of many trees and the robustness of the forest increase as the number of trees increase. Decision trees are built on random samples from the dataset. Predicted results from each tree are collected and the result with the greatest number of votes is selected. This technique is called Bagging. As it consists of multiple decision trees, it is quite slow in giving predictions.

- ***Naive Bayes***

It is a machine learning classifier that is based on the Bayes Theorem. It takes into account the notion of independence of predictors. Simply put, the features of a class are independent of each other. These features are still considered independently even if they are interdependent. This theorem is considered naïve, and the computation is also simplified because of the assumption.The Bayes theorem formula is as follows:

$$P(m|n) = \frac{P(n|m)\ P(m)}{P(n)} \qquad \textit{Eq. (4)}$$

Where,

- P(m|n) is the conditional probability also called Posterior probability of class m with the predictor n given.
- P(m) and P(n) are the fixed and predictor probabilities respectively.
- P(n|m) is the likelihood of predictor class n with class m given.

a. Ensembled Boosting Model (EBM) classifier

Ensemble methods are machine learning algorithms that build a set of classifiers and then use a weighted vote of their predictions to categorize fresh data points. It is a general meta-approach to machine learning that combines the predictions from different models to improve predictive performance. Bagging, stacking, and boosting are the three primary classes of ensemble learning algorithms, and it's critical to have a thorough understanding of each classifier. Boosting includes incrementally adding ensemble members that correct preceding prediction accuracy and produces a weighted average of the predictions.

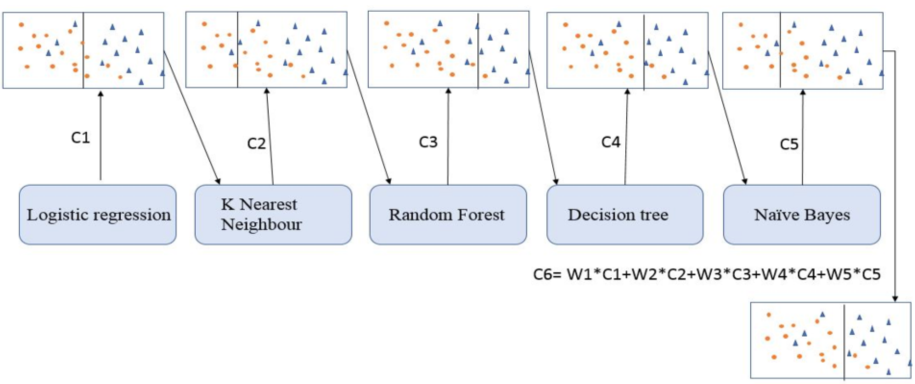

Figure 3. Ensembled Boosting Model (EBM) classifier Architecture diagram

The Logistic regression, KNN, Random Forest, Decision tree and Naïve Bayes is added sequentially to design a new Ensembled Boosting Model (EBM). The EBM starts classifying the data points initially by assigning equal weights to all the classifiers. After each model performance the weights are revised according to its classification accuracy. Misclassified data points are given more preference so that the next iteration concentrate more on those data points and weights are adjusted in order to classify them correctly.

EBM Algorithm

Begin Classification ()
Step 1: Assign Equal weights for Logistic regression, KNN, Random Forest, Decision tree and Naïve Bayes classifier.
Step 2: Aggregate all the misclassified data points by the classifiers in step 1.
Step 3: Improve the weights of the misclassified data points and perform the classification again with all the classifiers in step 1.
Step 4: If(classification accuracy < 97)
Repeat steps 2 to 4
Else
End Classification

The classification of data points using EBM classifier is accomplished using the *Eq. (5).*

$$C6 = W1 * C1 + W2 * C2 + W3 * C3 + W4 * C4 + W5 * C5 \qquad \textit{Eq. (5)}$$

WhereC1, C2, C3,and C5 are the base classifiers as mentioned in Step 1, W1, W2,..,and W5 are the weights of the base classifiers.

5. Results

The dataset has been trained with 6 algorithms - Logistic Regression, KNN, Random Forest, Decision Tree, Naive Bayes, and the proposed model EBM. The performance metrics like Accuracy, Mean Squared Error (MSE), F1-score, Precision, and Recall have been noted for each algorithm. The Performance measures are evaluated by computing the confusion matrix for each classifier. Using these measures, different performance assessment metrics are used to assess the results of classification performance. From *Table 1*, it can be inferred that the performance of Logistic regression and Naive Bayes is very poor when compared to the other algorithms. This is because Logistic regression is a prediction algorithm so it couldn't perform well with unlabeled data and Naive Bayes classification is not ideal for large datasets and data with too many features. The remaining three algorithms-

Table 1: Performance of the Algorithms

Algorithm	Mean Squared Error	Accuracy	F1-Score	Precision	Recall
Logistic Regression	9.445	0.257	0.068	0.109	0.11
KNN	1.34	0.969	0.967	0.967	0.967
Random Forest Classifier	1.255	0.97	0.969	0.973	0.965
Decision Tree Classifier	1.264	0.961	0.957	0.957	0.957
Naive Bayes	7.901	0.177	0.118	0.181	0.149
EBM	1.08	0.983	0.953	0.962	0.976

KNN, Random Forest, and Decision Tree show similar performance. The base classification algorithms have shown good results with accuracy rates greater than 96% and precision greater than 95%. Among the rest of the classification algorithms, EBM and Random Forest showsgood results with 98.3% and 97% accuracy.

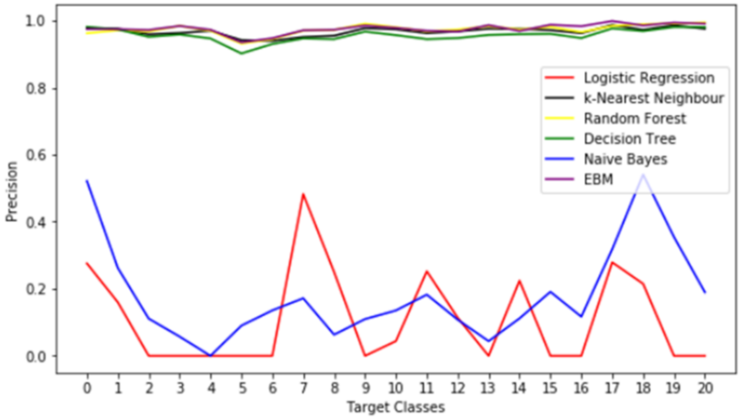

Figure 4. Precision trend of the algorithms

Figure 4 shows the precision trend of algorithms for each target class. It represents how precisely each algorithm was able to classify each label. From the Figure 4, it can be inferred that the precision line of the algorithms with high performance - KNN, Random Forest, Decision tree and EBM is quite stable when compared to Logistic regression and Naïve Bayes. The precision line of Logistic regression and Naïve Bayes has many fluctuations, it is only able to classify a few target classes with precision with a maximum of only 55%. The precision of the remaining classes has been between 10% to 20% or even less than 10%. The proposed EBM model shows the highest precision, and it can be concluded as best classifier.

6. Conclusion

The proposed work detects falls from the real-time data which would be obtained through a sensor attached to the bike. Related works only detect and predict falls in a closed environment with limited movements and only pertinent to one person, which has been overcome. The existing Machine Learning architectures perform with a maximum accuracy ranging from 90-95% only and with limitations of being applicable only in environments with a limited range of movements by elderly or unhealthy persons. The proposed EBM model for fall detection performs well with healthy and prime-aged persons and with an accuracy of 98.3%. In future work, this algorithm used for a real-time fall detection system can be extended which also predicts the falls based on kinematic data.

References

[1] H. Gaulrapp, A. Weber and B. Rosemeyer, ” Injuries in mountain biking,” Knee Surgery, Sports Traumatology, Arthroscopy, p. 48–53, 2000.

[2] F. M. Castro, R. Delgado-Esca˜no, J. R. C´ozar, M. J. Mar´ın-Jim´enez, N. Guil and E. Casilari, ”ACrossdataset Deep learning-based Classifier for People Fall Detection and Identification,” Computer Methodsand Programs in Biomedicine, pp. 1-21, 2019.

[3] L. Chen, R. Li, H. Zhang, L. Tian and N. Chen, ”Intelligent fall detection method based on accelerometerdata from a wrist-worn smart watch,” Measurement, pp. 215-226, 2019.

[4] E. Torti, A. Fontanella, M. Musci, N. Blago, D. Pau, F. Leporati and M. Piastra, ”Embedding RecurrentNeural Networks in Wearable Systems for Real-Time Fall Detection,” Microprocessors and Microsystems,pp. 1-10, 2019.

[5] E. Casilari-P´erez and F. Garc´ıa-Lagos, ”A comprehensive study on the use of artificial neural networksin wearable fall detection systems,” Expert Systems With Applications, pp. 1-26, 2019.

[6] A. Alarifi and A. Alwadain, ”Killer heuristic optimized convolution neural network-based fall detectionwith wearable IoT sensor devices,” Measurement, pp. 1-10, 2020.

[7] Y. Wu, Y. Su, R. Feng, N. Yu and X. Zang, ”Wearable-sensor-based pre-impact fall detection systemwith a hierarchical classifier,” Measurement, pp. 283-292, 2019.

[8] D. Sarabia-J´acome, R. Usach, C. Palau and M. Esteve, ”Highly-Efficient Fog-Based Deep Learning AalFall Detection System,” Internet of Things, pp. 1-19, 2020.

[9] D. V and L. S. R, ”Docker based Intelligent Fall Detection using Edge-Fog Infrastructure,” FACPapersOnLine., pp. 45-50, 2020.

[10] S. Nooruddin, M. M. Islam and F. A. Sharna, ”An IoT based Device-type Invariant Fall Detection System,”Internet of Things, pp. 1-24, 2019.

[11] S.Magesh, Indumathi, J, Radha Ram Mohan.S,Niveditha.V.R,Shanmuga Prabha(2020f): “Conceptsand Contributions of Edge Computing in Internet of Things (IoT): A Survey”, International Journalof Computer Networks and Applications (IJCNA),Volume 7, Issue 5,pages 146-154 . DOI:10.22247/ijcna/2020/203914.ISSN : 2395-0455

[12] R. Igual, C. Medrano and I. Plaza, ”A comparison of public datasets for acceleration-based fall detection,”Medical Engineering and Physics, pp. 870-878, 2015.

[13] R. M. Gibson, A. Amira, N. Ramzan, C.-d.-l.-H. P. and Z. Pervez, ”Multiple comparator classifier frameworkfor accelerometer-based fall detection and diagnostic,” Applied Soft Computing, p. 94–103, 2016.

[14] N. Kattukkaran, A. George and T. P. M. Haridas, ” Intelligent accident detection and alert system foremergency medical assistance,” International Conference on Computer Communication and Informatics(ICCCI), pp. 1-6, 2017.

[15] J. Baramy, P. Singh, A. Jadhav, K. Javir and M. S. Tarleka, ”Accident Detection & Alerting System,”International Journal of Technical Research and Applications, pp. 8-11, 2016.

[16] B. Barshan and A. T. O¨ zdemir, ”Detecting Falls with Wearable Sensors Using Machine Learning Techniques,”Sensors 2014, pp. 10691-10708, 2014.