



Article

Comparative Analysis of Real-Time Fall Detection Using Fuzzy Logic Web Services and Machine Learning

Bhavesh Pandya , Amir Pourabdollah * and Ahmad Lotfi

School of Science and Technology, Nottingham Trent University, Clifton Lane, Nottingham NG11 8NS, UK; bhavesh.pandya@ntu.ac.uk (B.P.); ahmad.lotfi@ntu.ac.uk (A.L.)

* Correspondence: amir.pourabdollah@ntu.ac.uk; Tel.: +44-115-848-3957

Received: 30 October 2020; Accepted: 1 December 2020; Published: 3 December 2020



Abstract: Falls are the main cause of susceptibility to severe injuries in many humans, especially for older adults aged 65 and over. Typically, falls are being unnoticed and interpreted as a mere inevitable accident. Various wearable fall warning devices have been created recently for older people. However, most of these devices are dependent on local data processing. Various algorithms are used in wearable sensors to track a real-time fall effectively, which focuses on fall detection via fuzzy-as-a-service based on IEEE 1855–2016, Java Fuzzy Markup Language (FML) and service-oriented architecture. Moreover, several approaches are used to detect a fall using machine learning techniques via human movement positional data to avert any accidents. For fuzzy logic web services, analysis is performed using wearable accelerometer and gyroscope sensors, whereas in machine learning techniques, k-NN, decision tree, random forest and extreme gradient boost are used to differentiate between a fall and non-fall. This study aims to carry out a comparative analysis of real-time fall detection using fuzzy logic web services and machine learning techniques and aims to determine which one is better for real-time fall detection. Research findings exhibit that the proposed fuzzy-as-a-service could easily differentiate between fall and non-fall occurrences in a real-time environment with an accuracy, sensitivity and specificity of 90%, 88.89% and 91.67%, respectively, while the random forest algorithm of machine learning achieved 99.19%, 98.53% and 99.63%, respectively.

Keywords: fall detection; fuzzy logic; machine learning; web services; real time; older adults

1. Introduction

Falls are one of the prime reasons for fatality among older adults and create a barrier for independent living in many cases. According to the statistics carried out by England and Wales [1], fall is the main reason behind death-related injuries among older adults aged 79 and over and the second leading reason for unintentional deaths among all ages. Early fall detection among elderly people is a thoughtful conundrum reported by the World Health Organization (WHO). Approximately 28–35% of people between 65 and 75 years and 32–45% over 70 years suffer from a fall [2]. Current research has shown that the incidences of falls among the older adults are on the rise; therefore, as a precautionary measure, a system that responds to an older adult's fall must be developed. It is projected that by 2030, the number of fall-induced accidents is estimated to increase up to 100% [2]. A fall detection system can be an assistive tool with the primary principle of informing fall incidents. One of the main design objectives of fall detection systems is to alert all types of fall incidences, especially those related to Activities of Daily Lives (ADL) [3,4] thereby aiming to lessen injuries to the spine, head or severe bone fractures. In certain countries, Personal Emergency Response System (PERS) facilities are developed for the elderly in case a fall occurs. Reports indicate that 80% of older

adults are unable to stand after a fall and fail to use PERS to seek assistance [5,6]. Therefore, a smart automated and precise fall detection system will be a substantial part of daily living environment for the elderly to rapidly detect and provide fast medical responses [7]. Various fall detection systems involve using analytical data from images, videos and audio as well as data from inertial sensors like accelerometers and gyroscopes [3,4,8].

Moreover, wearable sensors (i.e., accelerometers and gyroscopes) that stimulate human activity monitoring using a rule-dependent Fuzzy Logic System (FLS) have been demonstrated [9]. Different machine learning algorithms, such as k-NN, decision tree, random forest and extreme gradient boosting, and Artificial Neural Networks (ANNs) detect a fall using wearable sensors. However, there are few research studies that have been carried out to determine comparative analysis of real-time fall detection using fuzzy logic web services and machine learning for analysing the performance of a real-time fall detection system. In summary, the main contributions of this paper include:

- A comprehensive study on the performance of various fall detection machine learning techniques to detect a fall.
- To analyse the performance of online and offline fall detection techniques. Specifically, Fuzzy-as-a-service utilises an online real-time approach and machine learning as an offline approach.
- To evaluate the efficiency of using fuzzy versus non-fuzzy approaches in wearable sensors-based real-time applications.

The remainder of this paper is organised as follows. Section 2 focuses on review of related literature with respect to fuzzy logic web services and machine learning techniques. Section 3 explains in-depth the system architecture of a fall detection system in addition to the methodology using fuzzy logic web services and a modelling approach using machine learning techniques. Section 4 emphasises on the results and discussion. Section 5 concludes the study with future scope.

2. Review of Related Literature

A great deal of research has been carried out on fuzzy logic web services for real-time fall detection by means of wearable accelerometer and gyroscope sensors, which can be referred in detail in [10]. This paper is an extension of the authors' research work. Generally, the techniques employed for fall detection systems can be categorised into three groups, namely, wearable devices, ambience sensors and vision-based sensors [7,10,11].

- (a) **Wearable devices:** Herein, a subject is required to wear a device or a sensor embedded in his/her garment to track his/her posture and motion. Sensor data that is collected is used as motion signals to analyse different body movements [12,13]. A specific threshold-based sensor is used that triggers an alarm whenever the output of the sensor reaches a specific threshold. Recently, motion-based sensors have been integrated with commercially based smartphones [14]. The advantage of using such sensors is that they provide flexibility and portability; however, their false alarm rate is high.
- (b) **Ambient Sensors:** Using various sensors to record human-related data whenever a subject is in close proximity. The main idea is to identify changes in posture from standing to lying down whenever a fall occurs. The limitation of this type of sensor is that accuracy detection is limited, it is cost inefficient and false alarm occurrence is high.
- (c) **Vision-based sensors:** An indoor camera is used to monitor a single subject's movement. A variety of video-processing algorithms use fall characteristics to determine a fall occurrence such as Verso Vision. The limitation of this type of sensor is that they are immovable, and a subject is confined to a region covered by a camera.

A great deal of research has been carried out in machine learning for fall detection. Table 1 shows the taxonomy of fall detection. The system introduced in [15] is employed to classify the affected body

parts in a fall as several three-axis acceleration sensors are used to measure the properties of various positions. If the acceleration reaches the normal acceleration limit, the model can decide if there are fall accidents. The model transfers the sensor information to the computer via wireless transmissions for further analysis and assessment across different body parts. It employs the cognitive adjustment method to adapt the range of acceleration of different body parts in different movements. The sensors were added to six areas of the body to reduce people's indisposition.

Authors in [16] proposed that body angular velocity threshold checks should be included for detecting a fall with significantly improved standards. By including such standards, they were of the opinion that the accuracy improved from 82.72% without a gyroscope to 96.2% with a gyroscope. Because sensitivity and specificity are combined, optimising key levels by receiver operating characteristic analysis was a systematic strategy to simultaneously maximise sensitivity and specificity. A wireless, wearable sensor system comprising a three-axis gyroscope and an accelerometer were placed at the centre of the chest. Tests were conducted in a laboratory environment for 36 individuals with a total of 702 different moves. 50% of the dataset was used by the algorithm for fall detection, including significant sensor thresholds, and the remaining 50% of the dataset was used for evaluating the algorithm's sensitivity and specificity.

Table 1. Fall detection research taxonomy.

Technique	Sub-Category	Ref.	Year	Sensor Type	Performance		
					Accuracy	Sensitivity	Specificity
Wearable-based	Attached to the body	[15]	2011	Multiple tri-axial accelerometers	–	–	–
		[16]	2015	Accelerometer and gyroscope	–	96.3%	96.2%
	[17]	2013	Accelerometer and gyroscope	99.38%	100%	99.38%	
	Mobile phone-based	[18]	2011	Multiple cameras	–	99.7%	99.7%
		[19]	2008	Multiple cameras	100%	–	–
Classification algorithm-based	Machine learning	[20]	2013	Tri-axial accelerometer	100%	100%	100%
		[21]	2016	Tri-axial accelerometer	97.2%	91.7%	100%
		[22]	2015	Accelerometer and magnetometer	97.7%	99.3%	96%
		[23]	2014	Accelerometer	97.5%	97%	99%
		[24]	2017	Accelerometer and gyroscope	99.23%	99%	99.37%
		[23]	2014	Accelerometer and gyroscope	93.3%	–	–
	Threshold	[25]	2015	Tri-axial accelerometer and gyroscope	–	–	100%
		[16]	2015	Thri-axis accelerometer and gyroscope	90%	96.3%	96.2%
		[26]	2006	Bi-axial gyroscope	100%	100%	100%
		[17]	2013	Accelerometer and gyroscope	99.38%	100%	99.38%

Authors in [22] have proposed a support vector machine-based fall detection using low-cost Android smartphones. The proposed device had identified a total of 149 falls. The only event hidden in the fall was simulation with a syncope. With regard to ADL, there were merely six ADL incidents that were believed as fall events. Five imitations of collapsing on a chair and an imitation of lounging on a bed were the six ADL events incorrectly categorised as falls. Motorola Moto G (ver. 4.4.4) has been selected to fit the LIS3DH three-axis accelerometer with ST Microelectronics, an ultra-low power sensor with I2C and SPI interfaces with a selectable full scale of ± 16 g and competent of delivering data output rates to a maximum of 5 kHz. It also embedded a three-axis compass AK8963 that integrated a magnetic sensor that has a measuring range of ± 4900 T.

A smart phone accelerometer-based detection solution using a decision tree was proposed in [23]. The proposed approach could accurately predict fall events deprived of disrupting users with unnecessary false alarms, while offering the benefit of unaltering a user's habits because limited external sensors were needed. An unobtrusive fall detection system was proposed that made use of an amalgamation of information acquired from the classification of machine learning used in a state machine algorithm. Once the phone was in the user's belt or pocket, the data from the smartphone built-in accelerometer was screened unremittingly. The position of the user was monitored on identifying a fall occurrence, and short message service and electronic mail alerts were directed to a series of acquaintances.

Authors in [24] suggested a method that uses threshold-dependent ANNs. Threshold-driven feed-forward (FF) algorithm in ANN has the potential to improve the specificity and predictability

of a fall detection method from 96.25% to 99.37% and 97.69% to 99.23%, respectively. The intensity, however, was marginally decreased from 100% to 99%. This occurred on account of the prospect to give erroneous identification of ANN resulting in false negatives. Their research work aimed to build and create a modern waist-mounted fall detection device (Fallz 1.0) using an accelerometer and a gyroscope to monitor body orientation and rotation by motion sensors. Fallz 1.0 uses a motion sensor with an accelerometer and a gyroscope. Users wear the fall detection device. The body orientation and movement data comprising linear acceleration and angular velocity were transmitted to the algorithm for fall detection. The system alerted the fall and reported it to a caregiver or an observer to provide adequate assistance.

A web-based data language for FLS characterizations is the key criterion for implementing FaaS architecture. The current standard for this purpose is the IEEE-1855 (2016), also known as FML [27], an XML-based mark-up language allowing the human readable and hardware-independent definition of an FLS. FML and FML-compatible pieces of software such as JFML [28] are used as the basic design standard in this study, and the extensibility of this standard is a solution to architecture growth.

Soto-hidalgo et al. [28] proposed an open source Java library, i.e., JFML, which presented a complete implementation of the IEEE standard having a capacity to impart fuzzy systems as per different norms and programming. Arcos et al. [29] developed an interoperability unit to design and run FLS for embedded systems in JFML, particularly for Arduino boards. Moreover, they defined a transmission protocol between JFML and Arduino boards, which removed regulated computing capability proffered by embedded systems. Alcalá-Fdez et al. [30] expounded a Python wrapper for JFML (Py4JFML) that permits to use all JFML functionalities via Python programming language.

Considering the above review of related literature, this study aims to carry out a comparative analysis of real-time fall detection using fuzzy logic web services and machine learning and try to determine which one is better for a real-time fall detection.

3. Proposed System

Figure 1 indicates the general structure of the system proposed, containing three main steps, i.e., data acquisition, data processing and feature extraction of fuzzy data. Specifically, data from the accelerometer and gyroscope are continuously sampled and stored prior to the fall event.

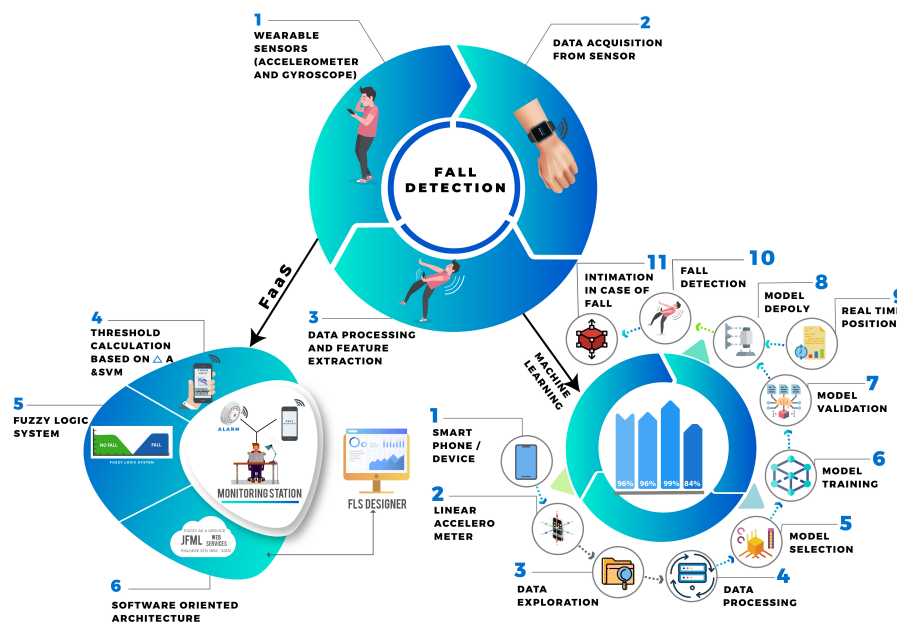


Figure 1. System architecture of a fall detection system.

3.1. Fuzzy Logic System

A normal fall can be typically viewed in three states: complete floating, state of impact and inactivity. The accelerometer and gyroscope then remove three functionalities. Checking to cause a sudden shift in acceleration in complete floating and impact states is one of the simplest methods for predicting the event of a fall. Like ADLs, hand clapping can easily trigger false alarms for wrist-wearing applications. During a fall, the flipping and spinning of hands might turn out to be critical. Two outputs can be obtained from the stage of data processing, Delta A (DA) and SVM values. Nevertheless, two simple criteria that distinguish falls for different people from all fall ADLs, particularly for the presence of constantly moving hands, are very difficult to determine. A fuzzy logic framework is implemented to solve the narrow thresholds provided by conventional Boolean logic 0 (fall) and 1 (non-fall) [26]. The fuzzy logical system takes the inputs over and takes a number of steps to generate vagueness-dependent output in an explicit testing. A standard fuzzy logic procedure is carried out in three phases, namely fuzzification, rule base and defuzzification. Fuzzification comprises fuzzifying each input values as a fuzzy membership function, fuzzy input set and compounding. Fuzzification turns signals into a fuzzy logic (low, medium and high) with degree of truth. The system is a dual input system which contains DA and SVM are the function of three members, i.e., low, medium and high DA values or low, medium, and high values, respectively. Memberships in a scheme are drawn and every membership is classified as turning points with different DA values choosing three turning points, i.e., 20, 45 and 90. Falls usually occur in 90, but some falls can occur in $<90^\circ$, and an exact DA cannot be determined by the wrist-worn sensor.

- (a) Fuzzify each input values as a function of fuzzy membership: Implement every necessary law to quantify the fuzzy output functions. To get “crisp” performance values, de-fuzzify the blurry output functions.
- (b) Fuzzy input set: SVM is the first input that contains three values, i.e., low, medium and high.
- (c) Compounding: Set a minimum angle, leading to a 45° fall, and see it as a medium angle. The lower and extreme angles are 20° and 90° respectively. The size of all 0° to 180° memberships were calculated by the minimum and maximum angles that the sensor may calculate. In that basis, if the angle is $>45^\circ$, then the accident is more likely to be called a collision.
- (d) Rule base: To perform this experiment, a total of nine rules were created for identifying whether it is a fall or not.
- (e) De-fuzzification: It is one of the main phases in the method that uses fuzzy logic to transform a fuzzy output set into a crisp value. As the input given to the system includes three values, i.e., low, medium and high, the output of the system offered three values, i.e., low, medium and high [10].

3.2. Methodology Using Fuzzy Logic Web Services

Following steps were used to obtain real-time fall detection using fuzzy logic web services:

(a) Data collection using wearable sensors: This study made use of an experimental dataset that is prepared with linear accelerometer sensors and gyroscope sensors installed in a smartphone, which is collected in real-time mode. Linear accelerometer sensors detected a fall or non-fall, whereas gyroscope sensors measured orientation and angular rotation. It consisted of time period and tri-axis human movement positioning data in addition to fall and non-fall labels. The labels contained information such as backside, forward and side falls (i.e., both left and right), normal walk and running. In total, data was gathered from six subjects, i.e., two subjects (20–25 years), 2 subjects (25–30 years), 1 subject (30–35 years) and 1 subject (60–65 years). The duration of each type of movement for the six subjects was 9–11 s. Data collection rate of the sensors was 419 Hz. For the five labels, each subject performed the aforementioned activities once. The classes were tested with different techniques. Moreover, it was determined that location data differed over time and was close to 0 in the event of a fall [10].

(b) Data exploration: Different features were assessed by means of several investigative techniques. Figure 2 shows the flow chart of the proposed fall detection system. Diagnostic data helps

in analysing nature as well as pattern of a fall as it provides an in-depth discernment. Positional data ensures continuous distribution of a fall; so, it is invigorated to analyse data that either follow normal distribution or not.

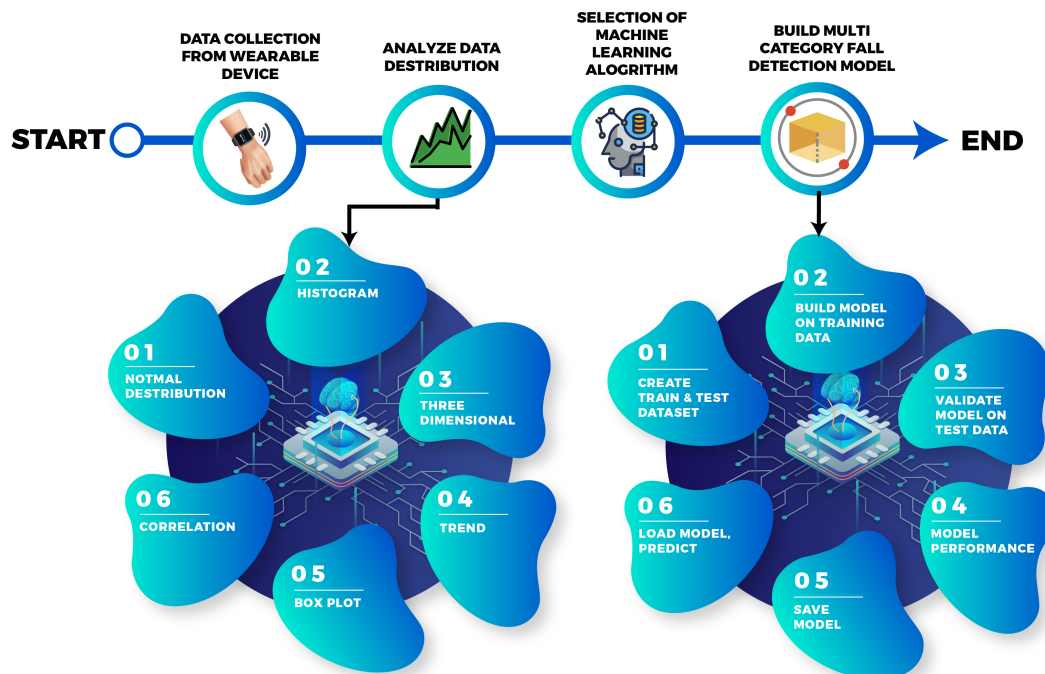


Figure 2. Flow chart of the proposed fall detection system.

From the Figures 3 and 4, it is evident that data frequency at position 0 is very high (for example, forward fall, backward fall and side fall), whereas in the case of a normal scenario data frequency position varies at different positions (for example, normal walk and running). After analysing data distribution by means of normal distribution and histogram plots, data visualization in three-dimensional (3D) space was needed. It is apparent that in case of a fall, data are being plotted based a fall's direction, whereas in the case of a normal scenario positional data are being concerted. 3D visualizations aid in envisioning data representation at a high dimension.

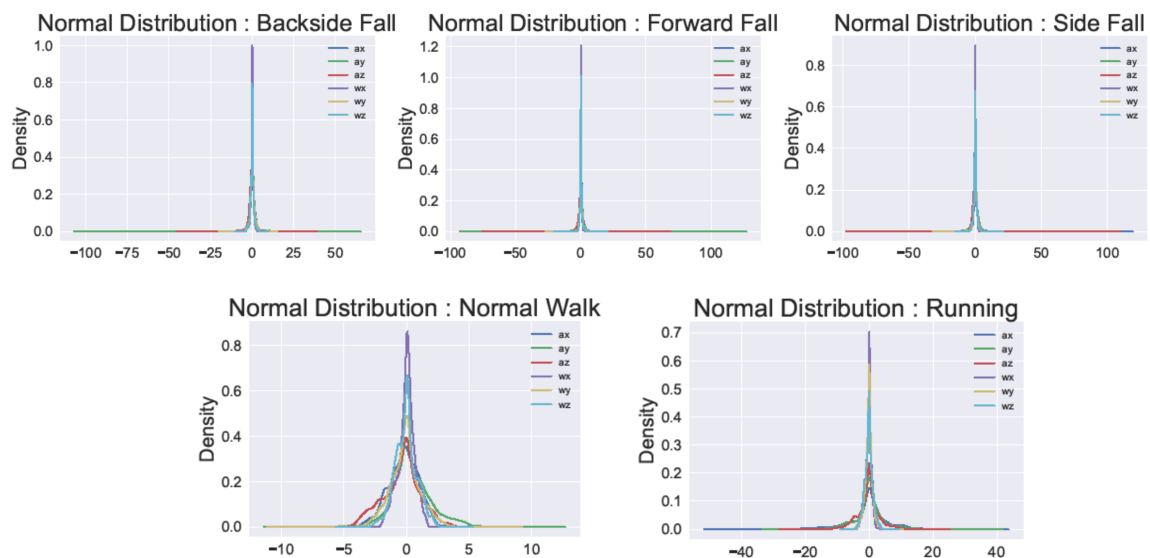


Figure 3. Normal distribution of the obtained accelerometer and gyroscope sensors.

Figure 3 shows a normal distribution in case of a fall. Sharp peaks at 0 can be observed, which indicate the nature of a fall. From the figure, it is evident that data tends to follow a normal distribution in case of a non-fall. For each class, data frequency in case of a fall and non-fall was checked, which was assessed via a histogram plot shown in Figure 4.

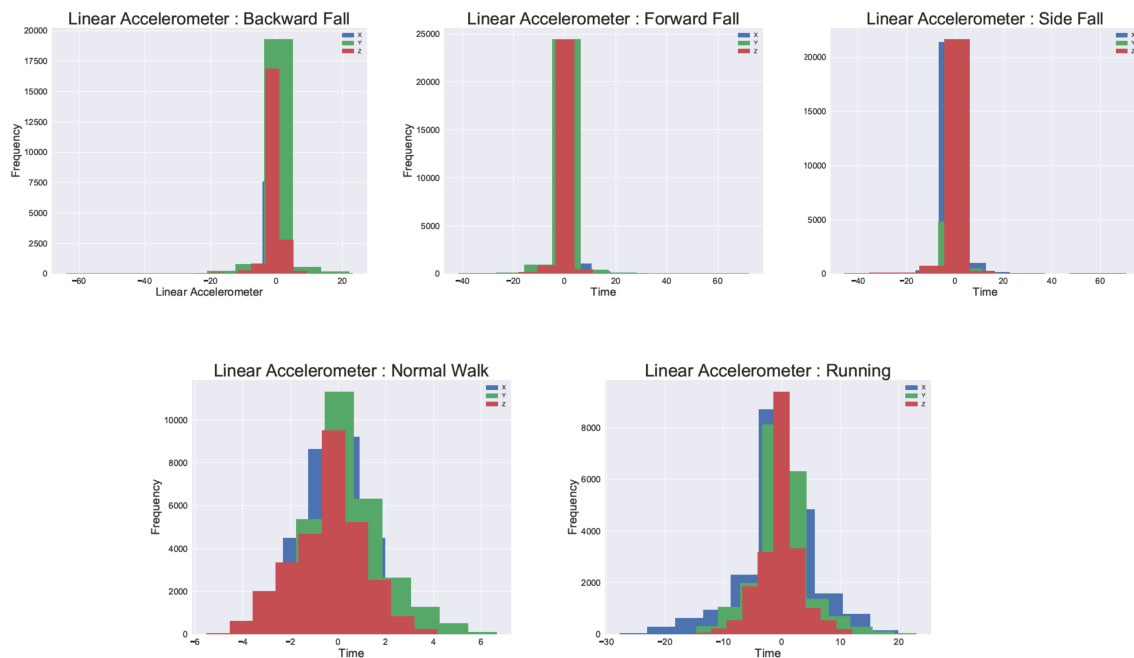


Figure 4. Histogram plots of the obtained accelerometer sensor.

Note that it is crucial to inspect trends in positional data with time. In case of a normal scenario, it was observed that positional data gets fluctuated with time and stay close to 0 in case of a fall. Box plots help in visualizing positional data range in every scenario. As it has already been observed in a normal distribution and trend plots, shows that in case of a fall, positional data ranges close to 0, whereas in the case of a normal situation, positional data varies intensely.

3.3. Modelling Approach Using Machine Learning Techniques

The generated dataset consists of positional data accompanied by response variables. A response variable is a multi-category variable with labels, namely backside, forward and side falls; normal walk and running. The dataset is also balanced with equal proportion of each response variable to circumvent the introduction of a bias factor in the model.

It is very important to identify the abstract features from the dataset in order to form a model that can provide the expected outcome. Hence, selecting a model is a crucial task, so that we have compared the performance of the selected machine learning model with model selection technique with the artificial neural network model.

Figure 2 shows the flow chart of the proposed fall detection system where modelling is done in three phases. In the first phase, the model selection technique is used to identify the best performing multi-class classification machine learning (ML) model, which outperforms the others. In the second phase, the selected ML model is used to build a fall detection system that can predict the response variable. In the final phase, the performance of the designed two-layered feed-forward artificial neural network is compared with the best selected ML model from the first phase.

In model selection phase, four classification algorithms are taken into account, i.e., k-NN, decision tree classifier, random forest classifier and extreme gradient boosting.

- (1) k-NN: Each object is graded by a majority vote of its neighbors, and the entity being allocated to the most common class of its nearest k neighbors. The purpose of considering k-NN model is because the nearby linear accelerometer or gyroscope sensor data points may form a specific pattern, which can be used to identify a fall or non-fall.
- (2) Decision tree classifier: In decision tree classification, branches represent independent variables and leaves represent class variables. The purpose of using decision tree classifier was to validate the effect of decision tree learning for prediction of a fall and non-fall.
- (3) Random forest classifier: It is a classification method for learning an ensemble. It is a set of decision trees from a randomly chosen training subset. It predicts the final class by aggregating the votes from various trees for decisions.
- (4) Extreme gradient boosting: is also an ensemble learning method and a decision tree-based algorithm where gradient descent optimization is used for minimizing errors to optimize parallel processing, tree pruning and the model's over-fitting.

A two-layered feed-forward ANN is also built to predict the fall. Here, the neural network can help us to learn the latent features among the data which may not be captured using discussed ML models. This neural network contains following layers:

- Input Layer: It has the same dimension as of input data. In our case, we have in total 6 features where 3 features represent linear accelerometer and the remaining 3 features represent the gyroscope data.
- Hidden Layer: There are 2 hidden layers, each having 600 neurons. Rectified Linear Unit (ReLU) is used as a non-linear activation function.
- Output Layer: As this is a multi-class classification problem and here we are aiming to predict 5 classes (backside fall, forward fall, side fall, normal walk, running), there will be 5 neurons in the output layer. Each neuron represent one class and at the end whichever neuron gets the highest probability, will be the final prediction.

The Softmax regression is a type of logistic regression that normalises an input value into a value vector that follows a distribution of probabilities with a total amount of up to 1.

$$S(y = j|z) = \frac{e^z}{\sum_{j=0}^k e^{z_k}} \quad (1)$$

where, we define the net input Z as,

$$z = \sum_{l=1}^m w_l x_l = W^T X \quad (2)$$

here, W is weight vector, X is feature vector [31]. Softmax function computes probability that training sample X belongs to class j that is forward fall, side fall, backside fall, normal walk, running, given the weight w and net input z . Table 2 shows the result analysis of ANN with 20 epochs offering a training accuracy of 93.28% and testing accuracy of 92.82% in 71 s.

Table 2. A summary of the experimental work in the training and testing stages using ANN with different number of epochs versus time-efficiency.

EPOCH	Training Accuracy (%)	Training Loss	Testing Accuracy (%)	Testing Loss	Time (s)
1	48.96%	1.2248	60.04%	0.9999	63
2	65.81%	0.8732	71.47%	0.7354	64
3	74.47%	0.6733	75.74%	0.6421	69
4	79.51%	0.5595	79.81%	0.5469	69
5	82.63%	0.4776	82.66%	0.4945	69
6	84.78%	0.429	83.92%	0.4504	71
7	86.45%	0.3886	86.10%	0.3956	70
8	87.70%	0.3595	88.22%	0.3526	70
9	88.62%	0.3411	87.41%	0.3768	69
10	89.56%	0.3181	88.38%	0.3335	69
11	90.18%	0.3037	89.44%	0.3251	70
12	90.61%	0.2918	90.94%	0.2795	72
13	91.22%	0.2834	90.15%	0.3195	68
14	91.51%	0.2815	91.33%	0.2997	71
15	91.81%	0.2772	91.12%	0.3189	69
16	92.30%	0.2616	91.28%	0.2996	72
17	92.67%	0.2511	91.96%	0.2839	69
18	92.88%	0.2447	91.30%	0.3092	70
19	92.98%	0.2472	92.01%	0.2851	71
20	93.28%	0.2386	92.82%	0.2514	71

4. Results and Discussion

In order to test the fuzzy based fall detection algorithm, there are four specific choices: True Positive (TP) shows that fall is accurately recognized, False Negative (FN) this means that fall is not noted, particularly where fall occurs in very slow motion, True Negative (TN) indicates that non-fall event categorised and, False Positive (FP) and True Negative (TN) non-fall event identified as a fall. In this study, 30 samples were collected from 6 participants, 18 of whom were enrolled. In the fall operation, 12 were held and 12 were held for non-fall behavior. The data set was divided into 70:30 of the train-test ratios. As shown in Table 3 the sensitivity outcome for fall activity is 88.89 percent, specificity result could reach 91.67 percent as shown in Table 1 and 90 percent is the total accuracy of fall and non-fall activities.

Table 3. Human activity monitoring result [10].

Predicted Condition		Results	
TP	16	TP + FN	18
FN	02		
TN	11	TN + FP	12
FP	01		
Sensitivity Accuracy	88.89%	Specificity	91.67%
			90%

Table 4 shows a comparison of different machine learning algorithms in terms of their average accuracy and standard deviation, and Table 5 shows the result analysis of Random Forest. In the model building stage, the dataset has been randomly divided into training and test sets with a 70:30 split ratio, respectively. A total of 97,320 samples were considered in our study, where 68,124 samples were used as a training data set and 29,196 samples were used as test data set. Table 2 shows feed-forward ANN which provides 92% accuracy. Although it is low compared with the RF method, it proves that ANN can also be a good alternative if data becomes more complex. It is shown that the RF method offers

a prediction accuracy of 98.53%. Moreover, the model can accurately predict the response variable because it achieves an overall sensitivity and specificity of 98.53% and 99.63%, respectively.

Table 4. Comparison of different machine learning algorithms.

Model Selection		
Algorithm	Average Accuracy	Standard Deviation
KNN	96.85%	0.20%
Decision Tree classifier	96.18%	0.14%
Random Forest	99.19%	0.10%
XGBoost	84.18%	0.53%

Table 5. Result analysis of Random Forest.

Random Forest							
No	Action	TP	TN	FP	FN	Sensitivity	Specificity
1	Backside Fall	5645	23,482	45	24	99.25%	99.48%
2	Forward Fall	5678	23,408	45	65	97.99%	99.43%
3	Side Fall	6003	23,114	33	46	98.33%	99.73%
4	Normal Walk	5866	23,270	39	21	98.94%	99.74%
5	Running	5806	23,312	36	42	98.13%	99.76%
-	Average	5799.6	23,317.2	39.6	39.6	98.53%	99.63%

5. Conclusions and Future Scope

Through successful implementation of the newly accepted IEEE-1855-2016 standard [9], this study presents analyses of real-time fall detection using a web-based service-oriented FLS architecture and machine learning-based methods. Fuzzy-as-a service [10] is capable of detecting a fall and a non-fall with an accuracy of 90%, whereas machine learning techniques can detect five classes of a fall, i.e., side fall, back fall, forward fall, walking and running with a maximum accuracy of 99.19% offered by the random forest technique. This paper confirms that fuzzy-as-a-service is capable to detect a fall from real-time data as it requires minimum hardware and software specification. Hence, this paper proposes a cloud-based fuzzy-as-a-service to generate real-time results. The main purpose of using fuzzy-as-a-service is to permit multipurpose delivery from clients to dedicated servers that perform complex computations, which are required for FLSs. Unequivocally, the use of virtualized cloud services provides the proposed system with elasticity. Reuse of existing data, balancing load amongst FLS devices and cost-efficiency are some of the advantages offered by FLS architecture. Future work could be to investigate the capabilities of the developed methodology in processing uncertain data for uncertain decision-making support scenarios in AAL through developing a service-oriented and web/cloud-based architecture for addressing the associated problem with uncertain data processing in AAL. This will involve exploring the applications, and possibly extending the standard web communication protocols for fuzzy logic systems, for example, IEEE 1855 and its associated software libraries. Future work can also evaluate the performance of the present wearable devices and ambient sensors such as Apple Watch Series 4, Bay Alarm Medical and Walabot.

Author Contributions: This project was conducted by B.P. as part of his PhD research at Nottingham Trent University. The project was supervised by A.P. The work report here is part of an ongoing research conducted by A.P. and A.L. B.P. has conducted the experiments and data analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Nottingham Trent University, Nottingham, UK.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADL	Activity Of Daily Living
ANN	Artificial Neural Network
FLS	Fuzzy Logic System
FML	Fuzzy Markup Language
ML	Machine Learning
PERS	Personal Emergency Response System
ReLU	Rectified Linear Unit

References

- Griffiths, C.; Rooney, C.; Brock, A. Leading causes of death in England and Wales—how should we group causes. *Health Stat. Q* **2005**, *28*, 6–17.
- World Health Organization; World Health Organization. Ageing; Life Course Unit. *WHO Global Report on Falls Prevention in Older Age*; World Health Organization: Geneva, Switzerland, 2008.
- Igual, R.; Medrano, C.; Plaza, I. Challenges, issues and trends in fall detection systems. *Biomed. Eng. Online* **2013**, *12*, 66. [[CrossRef](#)] [[PubMed](#)]
- Medrano, C.; Plaza, I.; Igual, R.; Sánchez, Á.; Castro, M. The effect of personalization on smartphone-based fall detectors. *Sensors* **2016**, *16*, 117. [[CrossRef](#)] [[PubMed](#)]
- Fleming, J.; Brayne, C. Inability to get up after falling, subsequent time on floor, and summoning help: Prospective cohort study in people over 90. *BMJ* **2008**, *337*, a2227. [[CrossRef](#)] [[PubMed](#)]
- Kunkel, D.; Pickering, R.M.; Ashburn, A.M. Comparison of retrospective interviews and prospective diaries to facilitate fall reports among people with stroke. *Age Ageing* **2011**, *40*, 277–280. [[CrossRef](#)] [[PubMed](#)]
- Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [[CrossRef](#)]
- He, J.; Bai, S.; Wang, X. An unobtrusive fall detection and alerting system based on Kalman filter and Bayes network classifier. *Sensors* **2017**, *17*, 1393. [[CrossRef](#)] [[PubMed](#)]
- Pandya, B.; Pourabdollah, A.; Lotfi, A. Fuzzy-as-a-Service for Real-Time Human Activity Recognition Using IEEE 1855–2016 Standard. In Proceedings of the International Conference on Fuzzy Systems, Glasgow, UK, 19–24 July 2020; submitted.
- Pandya, B.; Pourabdollah, A.; Lotfi, A. Fuzzy logic web services for real-time fall detection using wearable accelerometer and gyroscope sensors. In Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments, Corfu, Greece, 30 June–6 July 2020; pp. 1–7.
- Yu, X. Approaches and principles of fall detection for elderly and patient. In Proceedings of the HealthCom 2008-10th International Conference on e-health Networking, Applications and Services, Singapore, 7–9 July 2008; pp. 42–47.
- Nyan, M.; Tay, F.E.; Murugasu, E. A wearable system for pre-impact fall detection. *J. Biomech.* **2008**, *41*, 3475–3481. [[CrossRef](#)] [[PubMed](#)]
- Tong, L.; Chen, W.; Song, Q.; Ge, Y. A research on automatic human fall detection method based on wearable inertial force information acquisition system. In Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 19–23 December 2009; pp. 949–953.
- Casilari, E.; Oviedo-Jiménez, M.A. Automatic fall detection system based on the combined use of a smartphone and a smartwatch. *PLoS ONE* **2015**, *10*, e0140929. [[CrossRef](#)] [[PubMed](#)]
- Lai, C.F.; Chang, S.Y.; Chao, H.C.; Huang, Y.M. Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling. *IEEE Sens. J.* **2010**, *11*, 763–770. [[CrossRef](#)]
- Huynh, Q.T.; Nguyen, U.D.; Irazabal, L.B.; Ghassemian, N.; Tran, B.Q. Optimization of an accelerometer and gyroscope-based fall detection algorithm. *J. Sens.* **2015**. [[CrossRef](#)]
- Huynh, Q.T.; Nguyen, U.D.; Tran, S.V.; Nabili, A.; Tran, B.Q. Fall detection system using combination accelerometer and gyroscope. In Proceedings of the Second International I Conference on Advances in Electronic Devices and Circuits (EDC 2013), Kuala Lumpur, Malaysia, 6–7 June 2013. [[CrossRef](#)]

18. Auvinet, E.; Multon, F.; Saint-Arnaud, A.; Rousseau, J.; Meunier, J. Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *15*, 290–300. [[CrossRef](#)] [[PubMed](#)]
19. Auvinet, E.; Reveret, L.; St-Arnaud, A.; Rousseau, J.; Meunier, J. Fall detection using multiple cameras. In Proceedings of the 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vancouver, BC, Canada, 20–24 August 2008; pp. 2554–2557. [[CrossRef](#)]
20. Tong, L.; Song, Q.; Ge, Y.; Liu, M. HMM-based human fall detection and prediction method using tri-axial accelerometer. *IEEE Sens. J.* **2013**, *13*, 1849–1856. [[CrossRef](#)]
21. Cao, H.; Wu, S.; Zhou, Z.; Lin, C.C.; Yang, C.Y.; Lee, S.T.; Wu, C.T. A fall detection method based on acceleration data and hidden Markov model. In Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, China, 13–15 August 2016; pp. 684–689. [[CrossRef](#)]
22. Pierleoni, P.; Pernini, L.; Belli, A.; Palma, L.; Valenti, S.; Paniccia, M. SVM-based fall detection method for elderly people using Android low-cost smartphones. In Proceedings of the 2015 IEEE sensors applications symposium (SAS), Zadar, Croatia, 13–15 April 2015; pp. 1–5. [[CrossRef](#)]
23. Aguiar, B.; Rocha, T.; Silva, J.; Sousa, I. Accelerometer-based fall detection for smartphones. In Proceedings of the 2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA), Lisbon, Portugal, 11–12 June 2014; pp. 1–6. [[CrossRef](#)]
24. Yodpijit, N.; Sittiwanchai, T.; Jongprasithporn, M. The development of Artificial Neural Networks (ANN) for falls detection. In Proceedings of the 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 22–24 April 2017; pp. 547–550. [[CrossRef](#)]
25. Guo, H.W.; Hsieh, Y.T.; Huang, Y.S.; Chien, J.C.; Haraikawa, K.; Shieh, J.S. A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope. In Proceedings of the 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 28–30 November 2015; pp. 54–57. [[CrossRef](#)]
26. Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys.* **2008**, *30*, 84–90. [[CrossRef](#)] [[PubMed](#)]
27. Acampora, G.; Di Stefano, B.; Vitiello, A. IEEE 1855TM: The First IEEE Standard Sponsored by IEEE Computational Intelligence Society [Society Briefs]. *IEEE Comput. Intell. Mag.* **2016**, *11*, 4–6. [[CrossRef](#)]
28. Soto-Hidalgo, J.M.; Alonso, J.M.; Acampora, G.; Alcalá-Fdez, J. JFML: A java library to design fuzzy logic systems according to the IEEE std 1855-2016. *IEEE Access* **2018**, *6*, 54952–54964. [[CrossRef](#)]
29. Arcos, F.J.; Soto-Hidalgo, J.M.; Vitiello, A.; Acampora, G.; Alcalá-Fdez, J. Interoperability for Embedded Systems in JFML Software: An Arduino-based implementation. In Proceedings of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [[CrossRef](#)]
30. Alcalá-Fdez, J.; Alonso, J.M.; Castiello, C.; Mencar, C.; Soto-Hidalgo, J.M. Py4JFML: A Python wrapper for using the IEEE Std 1855-2016 through JFML. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 18–21 June 2019; pp. 1–6. [[CrossRef](#)]
31. Pias, T.S.; Eisenberg, D.; Islam, M.A. Vehicle Recognition Via Sensor Data From Smart Devices. In Proceedings of the 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 3–6 October 2019; pp. 96–99.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).